REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 01-11-2012	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 1-Apr-2008 - 31-Mar-2012

4. TITLE AND SUBTITLE
Modeling, Analysis, and Control of Swarming Agents in a Probabilistic Framework

5a. CONTRACT NUMBER
W911NF-08-1-0106

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
Devendra P. Garg, Gregory K. Fricke

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Duke University
130 Hudson Hall, Box 90271
Duke University
Durham, NC 27705 -

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
52488-CS.23

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This research effort focuses on the development of a unified, systematic, and formal approach to modeling and control of multi-agent systems, drawing inspiration from biological systems exhibiting swarm behavior. Previously developed works in the application of graph theory were further advanced and applied to a variety of different task-oriented multi-agent systems. New distributed on-board controllers were developed and implemented at the individual level, yielding collective results in exploration, perimeter detection and surveillance, and formation

15. SUBJECT TERMS
swarming robots, multi-robot control, heterogeneous agent formation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Devendra Garg
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 919-660-5330

Report Title

Modeling, Analysis, and Control of Swarming Agents in a Probabilistic Framework

ABSTRACT

This research effort focuses on the development of a unified, systematic, and formal approach to modeling and control of multi-agent systems, drawing inspiration from biological systems exhibiting swarm behavior. Previously developed works in the application of graph theory were further advanced and applied to a variety of different task-oriented multi-agent systems. New distributed on-board controllers were developed and implemented at the individual level, yielding collective results in exploration, perimeter detection and surveillance, and formation control that are indistinguishable from similar centrally controlled systems. This distributed control yields robustness and decreased complexity relative to centralized controllers. Simulations and experiments were conducted to validate the analytically obtained results. The major outcomes of this research have been published in a large number of peer-reviewed conference proceedings and scientific journal articles. The financial support provided by the Army Research Office to carry out this research has been duly acknowledged in these articles.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
08/25/2011	5.00	Dejan Lj. Milutinovic, Devendra P. Garg. Kalman Smoother Based Force Localization and Mapping Using Intravital Video Microscopy, Journal of Dynamic Systems, Measurement, and Control, (11 2010): 61503. doi: 10.1115/1.4002485
08/25/2011	8.00	D.P. Garg, V. Kumar, M. Kumar. Segregation of Heterogeneous Units in a Swarm of Robotic Agents, IEEE Transactions on Automatic Control, (03 2010): 743. doi: 10.1109/TAC.2010.2040494
11/01/2012	12.00	Guoxian Zhang, Gregory K. Fricke, Devendra P. Garg. Spill Detection and Perimeter Surveillance via Distributed Swarming Agents, IEEE/ASME Transactions on Mechatronics, (02 2013): 121. doi: 10.1109/TMECH.2011.2164578
11/01/2012	18.00	Reed F. Young, Devendra P. Garg. Modeling, simulation, and characterization of distributed multi-agent systems, Journal of Systemics, Cybernetics and Informatics, (02 2012): 73. doi:
TOTAL:		4

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper
TOTAL:	

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Garg, D., " Multi-Robot Cooperative Control: Challenges and Opportunities", presented as a keynote speech at the 13th IASTED International Conference on Control and Applications, Vancouver, British Columbia, Canada, June 1-3, 2011.

Number of Presentations: 1.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
08/25/2011	1.00	Gregory K. Fricke, Adam W. Caccavale, Devendra P. Garg. Mobile sensor frame mapping via vision and laser scan matching, 2010 3rd International Symposium on Resilient Control Systems (ISRCS). 2010/08/10 00:00:00, Idaho Falls, ID, USA. : ,
08/25/2011	2.00	Dejan Milutinovic, Devendra P. Garg. A Sampling Approach to Modeling and Control of a Large-Size Robot Population, ASME 2010 Dynamic Systems and Control Conference. 2011/09/13 00:00:00, Cambridge, Massachusetts, USA. : ,
08/25/2011	3.00	Guoxian Zhang, Devendra P. Garg, Greg Fricke. Hazardous Spill Perimeter Detection and Monitoring via Multiple Autonomous Mobile Robotic Agents, ASME 2010 Dynamic Systems and Control Conference. 2010/09/13 00:00:00, Cambridge, Massachusetts, USA. : ,
08/25/2011	4.00	Gregory K. Fricke, Guoxian Zhang, Adam Caccavale, Walter Li, Devendra P. Garg. An Intelligent Sensing Network of Distributed Swarming Agents for Perimeter Detection and Surveillance, ASME 2010 Dynamic Systems and Control Conference. 2010/09/13 00:00:00, Cambridge, Massachusetts, USA. : ,
08/25/2011	6.00	Devendra P. Garg, Reed F. Young. Modeling, Simulation, and Characterization of Distributed Multi-Agent Systems, The 2nd International Multi-Conference on Complexity, Informatics and Cybernetics: IMCIC 2011. 2011/03/27 00:00:00, . : ,
11/01/2012	13.00	Devendra P. Garg, Manish Kumar, Vijay Kumar. Self-sorting in a swarm of heterogeneous agents, 2008 American Control Conference (ACC '08). 2008/06/11 00:00:00, Seattle, WA. : ,
11/01/2012	14.00	Manish Kumar, Dejan Milutinovic, Devendra P. Garg. Role of stochasticity in self-organization of robotic swarms, 2008 American Control Conference (ACC '08). 2008/06/11 00:00:00, Seattle, WA. : ,
11/01/2012	15.00	Guoxian Zhang, Devendra P. Garg. Mobile Multi-Robot Control in Target Search and Retrieval, ASME 2008 Dynamic Systems and Control Conference. 2008/10/22 00:00:00, Ann Arbor, Michigan, USA. : ,
11/01/2012	16.00	Gregory K. Fricke, Devendra P. Garg, Dejan Milutinovic. Sensing and Estimation on a Modular Testbed for Swarm Robotics, ASME 2009 Dynamic Systems and Control Conference. 2009/10/12 00:00:00, Hollywood, California, USA. : ,
11/01/2012	17.00	Manish Kumar, Devendra P. Garg. Aggregation of heterogeneous units in a swarm of robotic agents, 2011 4th International Symposium on Resilient Control Systems (ISRCS). 2011/08/09 00:00:00, Boise, ID, USA. : ,
11/01/2012	19.00	Gregory K. Fricke, Devendra P. Garg. Discrimination and tracking of individual agents in a swarm of robots , American Control Conference (ACC), 2010. 2010/06/30 00:00:00, . : ,
11/01/2012	21.00	Kevin M. Lieberman, Gregory K. Fricke, Devendra P. Garg. Decentralized Control of Multi-Agent Escort Formation via Morse Potential Function, ASME Dynamic Systems and Control Conference. 2012/10/17 00:00:00, . : ,

11/01/2012 20.00 Gregory K. Fricke, Kevin M. Lieberman, Devendra P. Garg. Swarm Formations Under Nonholonomic and Numerosity Constraints,
ASME Dynamic Systems and Control Conference. 2012/10/17 00:00:00, . : ,

11/01/2012 22.00 Gregory K. Fricke, Dejan Milutinovic, Devendra P. Garg. Robotic pose estimation via an adaptive Kalman Filter using state-varying noise,
IASTED 2009 Robotics and Automation Conference. 2009/11/02 00:00:00, . : ,

TOTAL: **14**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

08/25/2011 9.00 Gregory Fricke, Devendra Garg, Guoxian Zhang. Spill perimeter detection and Surveillance via distributed swarming agents,
IEEE Transactions on Mechatronics (08 2011)

08/25/2011 10.00 Gregory Fricke, Devendra Garg, Bruce Rogers. On the stability of swarm consensus under noisy control,
Proceedings of the 2011 Dynamic Systems and Control Conference (08 2011)

08/25/2011 11.00 Gregory Fricke, Devendra Garg, Bruce Rogers. Aggregation and Rendezvous in an Unbounded Domain without a Shared Coordinate System,
Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference (03 2011)

TOTAL: **3**

Number of Manuscripts:

Books

Received Paper

TOTAL:

Patents Submitted

No patent applications were submitted during the period of this research project.

Patents Awarded

No patent titles were awarded during the period of this research project.

Awards

Honors:

Plenary Speaker, 2011 13th IASTED International Conference on Control and Applications, Vancouver, British Columbia, Canada, June 1-3, 2011.

Chief Guest, 2009 Convocation Prize Distribution Function, Indian Institute of Technology, Roorkee, India, November 14, 2009.

Member, International Advisory Board, 3rd International Conference on Smart Materials, Structures and Systems, Acireale, Sicily, Italy, June 8-13, 2008.

Member, Program Committee, 2008 International Symposium on Flexible Automation(ISFA) (Formerly, Japan-USA Symposium on Flexible Automation), Atlanta, GA, June 23-26, 2008.

Member, International Advisory Committee, 2008 IEEE international Conference on Automation and Logistics, Qingdao, China, September 1-3, 2008.

Member, International Program Committee, 3rd International Conference on Positioning Technology, Shizuoka, Japan, November 26-28, 2008.

Member, International Advisory Board, 2008 International Conference on Information & Automation for Sustainability, Sri Lanka, December, 2008.

Chairman, Data Fusion Track, 3rd IEEE International Symposium on Resilient Control Systems, Idaho Falls, ID, August 10-12, 2010.

Chairman, Data Fusion Track, 4th IEEE International Symposium on Resilient Control Systems, Boise, ID, August 09-11, 2011.

Chairman, Data Fusion Track, 5th IEEE International Symposium on Resilient Control Systems, Salt Lake City, UT, August 14-16, 2012.

Awards:

Yasundo Takahashi Education Award, Dynamic Systems and Control Division of ASME, Arlington, VA, October 2011.

Certificate of Appreciation, U.S. Army Research Office, Department of Defense, May 2011.

Capers and Marion McDonald Award for Excellence in Mentoring and Advising, Edmund T. Pratt School of Engineering, Duke University, April 2011.

IIT/Roorkee Distinguished Alumnus Award, IIT/Roorkee, India, 2009.

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Gregory K. Fricke	1.00	
Reed F. Young	1.00	
Guoxian Zhang	0.00	
FTE Equivalent:	**2.00**	
Total Number:	**3**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Dejan Milutinovic	0.00
Manish Kumar	0.00
FTE Equivalent:	**0.00**
Total Number:	**2**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Devendra P. Garg	0.16	
FTE Equivalent:	**0.16**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Timothy Burns	0.05	Mechanical Engineering
S. Walter Li	0.00	Mechanical Engineering
Adam Caccavale	0.00	Mechanical Engineering
FTE Equivalent:	**0.05**	
Total Number:	**3**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	3.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:......	3.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:......	1.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):......	3.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:......	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:......	1.00

Names of Personnel receiving masters degrees

NAME	
Gregory K. Fricke	
Total Number:	**1**

Names of personnel receiving PHDs

NAME	
Reed F. Young	
Total Number:	**1**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

Please see the attached document.

Technology Transfer

Modeling, Analysis and Control of Swarming Agents in a Probabilistic Framework

Devendra P. Garg
Gregory K. Fricke

November 1, 2012

Foreword

In this research effort, the investigators sought a unified, systematic, and formal approach to modeling and control of multi-agent systems, drawing inspiration from biological systems exhibiting swarm behavior. Previously developed works in the application of graph theory were furthered and applied to a variety of different task-oriented multi-agent systems. New distributed on-board controllers were developed and implemented at the individual level, yielding collective results in exploration, perimeter detection and surveillance, and formation control that are indistinguishable from similar centrally controlled systems. This distributed control yields robustness and decreased complexity relative to centralized controllers. Simulations and experiments were conducted to validate the analytical results. The results of this research have been published in a variety of conference proceedings and journal articals, which are reproduced in the appendix. This document serves to report the research conducted with the support of the Army Research Office under grant number W911NF-08-0106.

Contents

List of Figures

Chapter 1

Statement of Problem

Swarm control has gained considerable attention recently among the robotics research community. The attention is primarily motivated by the existence of very robust biological counterparts such as swarming in ants, flocking of birds, schooling in fishes, and even cellular interactions. The primary feature of these systems that has attracted researchers is that the intelligence associated with an individual agent (e.g. ant or bird) is very primitive, and it utilizes interactions at local level to arrive at very simple decisions. This behavior at the local level emerges into a group behavior that appears to be very robust and complex. This observation in biological systems has led the interested scientists and engineers to investigate multiple agent cooperative controls problem using a bottom-up approach.

1.1 Motivation

Several of the future military missions are likely to depend on a multitude of interconnected autonomous vehicles working cooperatively in a hostile, dynamic, and uncertain environment and equipped with sensing and communication capabilities. A few of the potential applications of such multi-vehicle systems include cooperative search and rescue operation, surveillance, reconnaissance, and perimeter protection. Advances in communication and sensing technologies, and in computing resources have made it possible to coordinate the movement of several autonomous vehicles working cooperatively to achieve a specified mission. However, such systems present considerable technological challenges in terms of management and control of large numbers of vehicles, communication between vehicles and with a central supervisory controller, and estimation from sensory sources. This, added to the fact that most military applications take place in very harsh, dynamic, uncertain, and hostile environments, necessitates the development of a robust and fault tolerant system that can operate efficiently in real time.

1.2 Research Goals

There are several issues that arise while designing such a system of autonomously acting vehicles working together as a swarm. Most critical are the stability, convergence, and robustness characteristics of these systems operating in dynamic and hostile environments. The scalability of the concepts and algorithms is important since a large number of autonomous agents constitute the overall system.

Another important issue is cohesiveness. The system should remain connected and algorithms should not allow fragmentation in groups of agents. The algorithms should include inbuilt mechanisms for obstacle avoidance and for carrying out split/rejoin maneuvers. Swarm systems would use sensors and communication devices for maintaining the interconnections and exchanging information. Such interconnections are liable to be broken. The coordination algorithms should be robust and tolerant to such sporadic broken connections. Finally, the group of vehicles should be able to collectively organize themselves in an optimal manner to achieve a specified objective or mission.

In view of the factors mentioned above, an in-depth research effort was conducted towards accomplishing the following goals:

- Development of a comprehensive and unified analytical approach to manage, control, and coordinate a large number of autonomous mobile agents.
- Formulation of graph theoretic techniques that draw inspiration from complex network theory to accurately model the connection uncertainties in these complicated systems of interconnected agents in dynamic environments via probabilistic and random graph theory, and derivation of novel results for various quantities of significance such as connectedness, diameter, and cluster size.
- Establishment of novel control methodologies motivated from self-organizing behaviors found in biological systems that can be adapted to control and coordinate a large number of autonomous agents working together to achieve a defined objective.
- Development of innovative methods at the intersection of systems and control theory and graph theory to analyze the stability, robustness, convergence properties, and cohesion of such interconnected agents.
- Design of experiments using a testbed consisting of several mobile agents with embedded processing, sensing, and communication capabilities.
- Extensive verification of proposed algorithms via computer simulations, as well as via laboratory experiments, representing a variety of cooperative control applications such as formation of various shapes, cooperative data gathering and area coverage, and perimeter protection.

Each of these objectives has been investigated, leading to novel control methodologies and practical solutions. In some cases, full implementation was not possible due to practical issues relating to available equipment. In these cases, the limitations were carefully observed, and workarounds proposed.

The next chapter includes summarized results of each component of our research. The appendix includes the collection of peer-reviewed journal papers and conference papers in which our research is described in much greater detail.

Chapter 2

Summary of Results

The work done under and outcomes of this research have been reported in several journal articles [1–4] and peer-reviewed conference proceedings [5–20].

2.1 Self-sorting of Heterogeneous Swarm Populations

Sorting and pattern formation of heterogeneous units are self-organized behaviors seen in many biological systems. One of the best examples of such systems is a system of biological cells of heterogeneous types that has the ability to self-organize into specific formations, form different types of organs and, ultimately, develop into a living organism.

Earlier research in this area has indicated that such pattern formation and self-sorting behaviors in biological cells and tissues are made possible because of difference in the adhesivity between different types of cells or tissues. Inspired by this differential adhesivity model, our research explored a decentralized approach based on differential artificial potential to achieve the self-sorting and pattern formation behaviors by a swarm of heterogeneous robotic agents. The method is based on the proposition that agents of different types experience different magnitude of potential while they are interacting with agents of different types.

Our research examined the stability analysis of a population of agents in the Lyapunov sense, and has laid down an analytical foundation for the synthesis of controllers for aggregation and segregation with the artificial potential function framework. Specifically, conditions for the synthesis of controllers for aggregation and segregation was analyzed in one and two dimensional space. Extensive simulation studies and numerical analysis suggest that the proposed methods for segregation or aggregation would always lead a population of heterogeneous agents closer to the sorted (i.e., segregated) or aggregated configuration, respectively. Details of this research are reported in [5, 16] and reproduced in Appendix A in Section A.2.1 and Section A.2.12 respectively.

2.2 The Role of Stochasticity for Swarm Cohesion

This component of the research focused on the effectiveness of designed random behavior in self-organization of swarm of robotic agents. Inspired by the self-organization observed in biological cells and the role played by random forces in providing robustness in cell self-organization, we investigated the possibility of designing a decentralized controller for a swarm of agents in which the stochastic process is included. Flocking was considered as a self-organizing behavior example to validate our findings. The controller is designed in the framework of Lyapunov function, and it is based on the artificial potential due to interactions among agents. Our analysis shows that the flocking behavior of the swarm is improved and is more robust when the stochastic process is included in the agent controller.

Flocking is a self-organized behavior in which agents, initially distributed over the operating space, group together and organize into a specific formation. An example of this kind of flocking is the formation of germinal centers inside the lymph node [21]. Flocking has been extensively studied in multi-agent literature. Reynolds [22] has been able to reproduce, in his computer models, behaviors representing flocking in birds and schooling in fish using simple rules based on local interactions among agents. Drawing inspiration from Reynolds approach, many researchers have focused on designing a decentralized controller for achieving flocking behavior [23–25]. A control system based on the methods presented in these references should be able to yield a single flock of agents based on local information. However, if the local information can only be collected over a finite range, it can lead to the formation of more than one flock.

This research focused on scenarios in which local interaction leads to the fragmentation of groups and investigates how the introduction of random processes in an agent controller helps to eliminate or alleviate the problem of fragmentation. We consider flocking in the Lyapunov function framework. The Lyapunov function for the controller design of a multi-agent system is based on the artificial potential function of agent interactions [26, 27]. Inspired by the role played by the random fluctuation in driving self-organizing processes in cells, we include a random process in the decentralized controller for an agent swarm. Consequently, we analyze the swarm selforganization using the stochastic Lyapunov function. We show that our proposed decentralized controller provides the robust flocking of the swarm of agents.

We investigated the possibility of designing a decentralized controller for a swarm of agents in which a stochastic process is included. We considered the flocking behavior of a swarm and described the previously considered deterministic Lyapunov function controller design based on the artificial potential of interactions among agents. The deterministic design may not lead to desired swarm behaviors because of the existence of agent configurations in which the total potential of swarm interactions has local minima. Comparisons of the results using determinstic and stochastic controllers are shown in Figure 2.1,

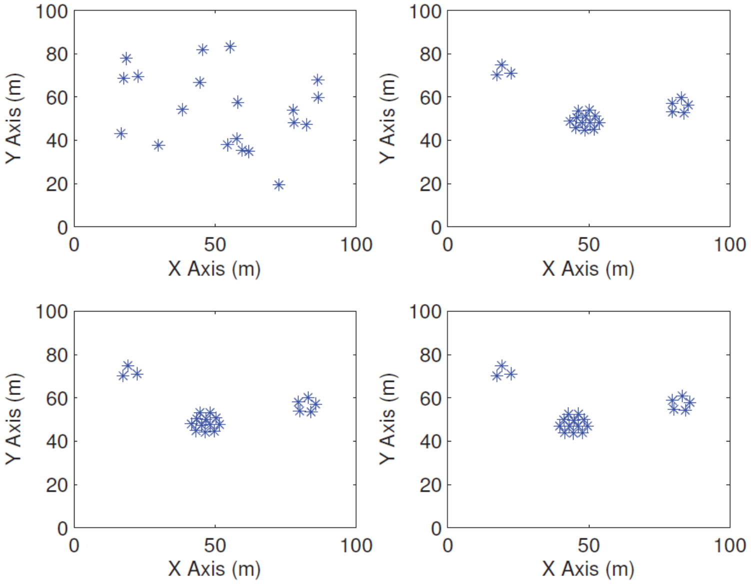


Figure 2.1: Agent clustering under deterministic controller.

Figure 2.2, and Figure 2.3.

The design we considered is an extension of the deterministic one. We used the same Lyapunov function. However, due to the introduced stochastic process, the Lyapunov function is also a stochastic process. Consequently, this controller provides a means of escaping from the local minima of the total potential of the swarm interactions. The intensity of the included stochastic process is of great significance. If its intensity is small, then the swarm behavior is the same as if controlled by the deterministic controller. If the intensity is high, then the flocking behavior cannot be established because the swarm can even escape from the region around the global minimum. Only if the intensity of the stochastic process lies within some intermediate range, does the swarm escape the local minima and randomly explores configurations, which can ultimately lead the swarm towards configurations close to the global minimum of the total potential of interactions. The drawback of this design is that even in the global minimum configuration, the stochastic process forces the swarm to search for a better configuration. Tuning the intensity of the stochastic process of the controller is identical to consideration that appears in the design of the realistic model of cell behavior, or to the problem of providing physiological conditions for self-organization in biological cells.

The performance of the proposed controller was illustrated by the derivation of the emergent swarm behavior and an extensive simulation study.The performance of the controller with the random process term was compared to the performance of the deterministic controller. The simulation results suggest

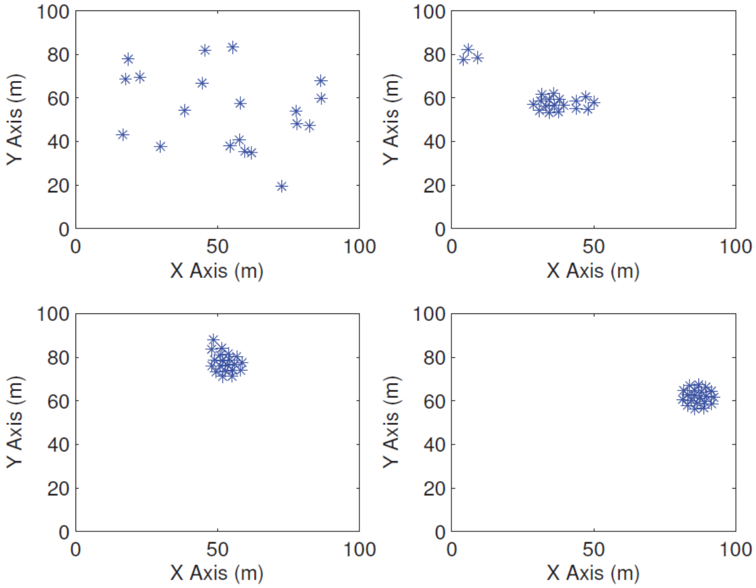


Figure 2.2: Agent clustering under stochastic controller.

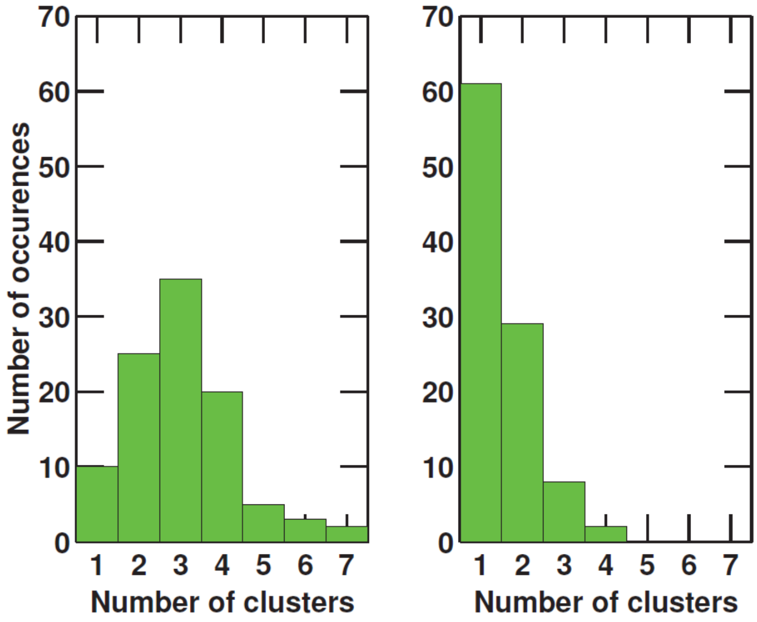


Figure 2.3: Cluster frequency with and without stochastic component.

that the inclusion of the random process in the controller can certainly improve the performance of the robotic swarm in achieving the flocking behavior. Additional details of this research can be found in [6], reproduced in Appendix A in Section A.2.2.

2.3 Kalman Smoother Based Force Localization and Mapping

Motility is an important property of immune system cells. It provides cells with the ability to perform their function not only at the right time but also at the right place. In this research, we introduce the problem of modeling and estimating an effective force field directing cell movement by the analysis of intravital video microscopy. A computational approach is proposed for solving this problem without dealing with a parametrized spatial model of the field in order to avoid potential errors due to inaccurate spatial model assumptions.

We consider the dynamics of cells similar to the dynamics of distributed agents typically used in the field of swarm robotics. The method utilizes a fixed-interval Kalman filter based smoother. Its application results in a map giving the intensity and direction of the effective force field. The results show that real-time video images are a source of data, enabling us to visualize intriguing spatiotemporal phenomena inside immune system organs. The proposed approach can fill the existing gap between contemporary technology and quantitative data analyses present in the field of biosystems.

We introduce the problem of estimating the effective force field influencing cell motility based on intravital video microscopy. We name this problem force localization and mapping, or FLAM. Considering an individual cell as an agent, we find this problem similar to a well-studied problem in robotics, the so-called simultaneous localization and mapping, or SLAM. In both problems, the estimation is based on the individual agents motility model and the outcome is a map that visualizes the spatial structure within the agents environment.

In this work, we have developed an approach for the estimation of the force field influencing cell motility. For this approach, we used the optimal smoother estimator of the forces and formed a map that visualizes the force field based on its estimations. We did not impose any analytical constraints on the force field. We introduced a spatial correlation between the forces by integrating estimations over a grid of rectangular regions. Consequently, the force field is estimated and visualized by the average forces assigned to each rectangular region. Using the proposed method, we are confident to estimate the intensity and direction of the major constant component of the force field. We also find that the directions of the force field are satisfactorily identified.

Our results show that real-time video images are sources of data, enabling visualization, which can be used for studying intriguing spatiotemporal phenomena inside immune system organs. We hope that our approach and similar approaches can fill the existing gap between contemporary technology and quantitative data analyses present in the field. Additional details of this research can be found in [2], reproduced in Appendix A in Section A.1.2.

2.4 A Sampling Approach to Modeling and Control of a Large Size Robot Population

Motivated by the close relation between estimation and control problems, we explore the possibility to utilize stochastic sampling for computing the optimal control for a large-size robot population. We assume that the individual robot state is composed of discrete and continuous components, while the population is controlled in a probability space. Utilizing a stochastic process, we can compute the state probability density function evolution, as well as use the stochastic process samples to evaluate the Hamiltonian defining the optimal control.

In this research area, we consider a problem in which the presence of a large-size robot population in a desired region of operating space is maximized. This problem is formulated in a hybrid system framework [28]. Its solution, based on the minimum principle for partial differential equations, is presented in [29, 30], and it is solved numerically when the presence of the robots is maximized along one dimension (1D).

The Hamiltonian, which defines the optimal control, includes integral terms that depend on the solution of a system of partial dirrerential equations (PDE). This system of PDEs is in general difficult to evaluate and the numerical evaluation of integrals is prone to errors. However, we recognize that the problem solution can be simplified and propose to use samples of the stochastic processes to evaluate the Hamiltonian components from the expected values of the adjust state distribution.

The benefit of using a solution based on sampling, i.e., computational statistical methods, is that control problems in robotics could be solved faster. This possibility also depends on the ability to implement sampling and computations with samples into the processor computing the control.

We considered a large-size robot population control problem that had been previously formulated and solved in a probability space utilizing systems of PDEs. Solving these PDEs is computationally expensive; therefore, having in mind that the PDEs are in close connection with the stochastic process to be controlled, we explore an opportunity to utilize the stochastic process samples to compute the control.

This work resulted in an algorithm for generating the stochastic process that can be used to propagate the state PDF of the robot population. We show that the algorithm predicts exactly the state PDF evolution and we derive expression for the Hamiltonian evaluation which involves the stochastic process samples. The Hamiltonian evaluated in this way can be used in iterations computing the optimal control as if it was computed based on the PDE system solutions. We also notice that the cost function resulting from utilizing stochastic processes has smaller values than the cost function computed based on PDEs. This means that the evaluations involving the stochastic process samples are closer to the true values. Additionally, the time required to compute the control for a very large population with this stochastic method was reduced significantly: the average time required was a factor of 4.5 less than that required by the numerical PDE solver.

In summary, we can conclude that utilizing stochastic processes for computing control of multi-robot systems considering discrete, as well as continuous robot states is possible. By embedding stochastic process generators into analog circuits and utilizing them in dedicated processors for computing control, complex stochastic optimal control problems can be solved efficiently and potentially exploited for real-time multi-robot systems control. Additional details of this research can be found in [14], reproduced in Appendix A in Section A.2.10.

2.5 Mobile Sensor Frame Mapping via Vision and Laser Scan Matching

Cooperative mobile robots must have knowledge of their positions relative to the group in which they are operating. Common on-board sensors such as laser rangefinders may be used to detect and track other robots with high precision, though limited feature recognition and susceptibility to occlusion reduces the efficacy of this solution alone. Matching multiple robots laser scans can overcome some of these issues, but requires extensive memory usage and large communication bandwidth. Overhead imaging systems may also be utilized, though sensor nonlinearities, field of view restrictions, and data latency limit such usage. A data fusion method was developed to dynamically evaluate mobile robots' positions by matching laser scan data to overhead image data.

We developed an architecture for relative localization of members of a robot collection by the fusion of various sensor data sets. This method is based on a combination of multiple laser rangefinders making in-plane measurements of the environment (which includes neighboring robots) with one or more optical overhead images taken by a supervisory observer.

The ultimate goal of this matching is to provide a method of unifying multiple sensor frames into a single relative frame, including correction for rotation, translation, and scaling. As of this writing, the problem of skewing or stretching has not yet been addressed.

Figure 2.4 shows an on-board laser data superimposed onto sketchup of lab layout, including obstacles and other robots. The grey circle indicates a typical field of view for a camera field-of-view on one of the aerial vehicles.

Preliminary results are promising, though the parameter space has yet to be explored for robustness, processing efficiency and the scalability of this problem. If this method is to be used with a collective or swarm robotic system, the processing requirements must be fixed (or nearly so) with increasing numbers of robots, requiring that each individual robot must do as much processing as possible. Continuing work focuses on more efficient methods, such as the search

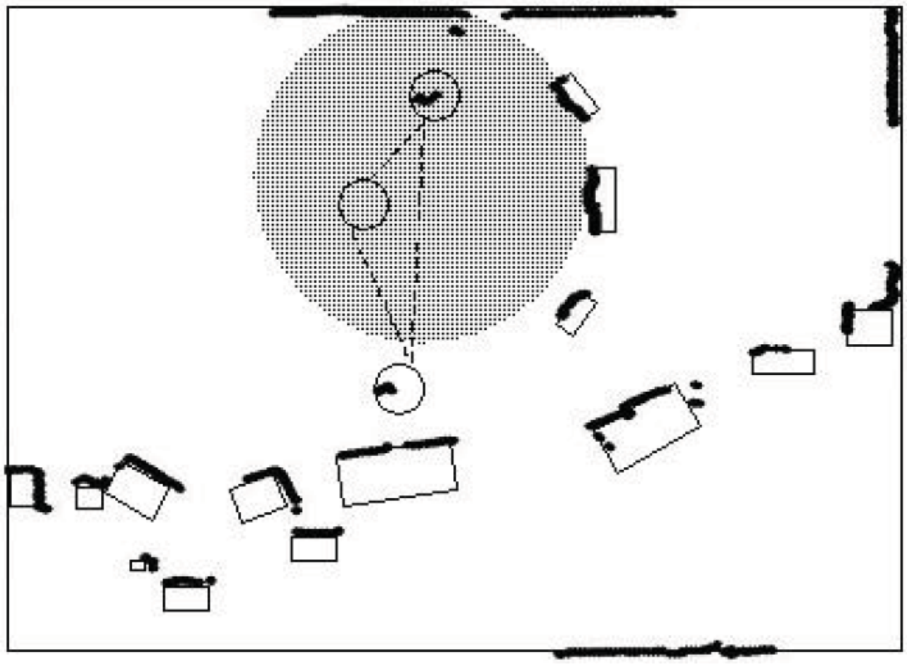


Figure 2.4: Laser data superimposed on arena schematic.

algorithms in [31] and the use of graph topology for image processing [32].

Additional details of this research can be found in [11], reproduced in Appendix A in Section A.2.7.

2.6 Unstructured Formation Control in Detection of Hazardous Perimeters

The problem of perimeter detection and monitoring has a variety of applications. In this aspect of our research, inspired by flocking agents, a hybrid control law with finite states was explored for a group of autonomous robotic agents whose purpose was to search for, detect, and track a hazardous spill in an unknown environment.

In the system, each robotic agent was allowed to be in one of three states: searching, pursuing, and tracking. The agents were prioritized based on their states, and a potential field was constructed for agents in each state. For an agent in the tracking state, the agents location and velocity as well as those of its closest leading and trailing agents were utilized to control its movement.

Each agent was assumed to have limited and specific sensing and communication ranges. It was shown that the group can successfully detect the spill and track its boundary when the spills location and size are changing. The conver-

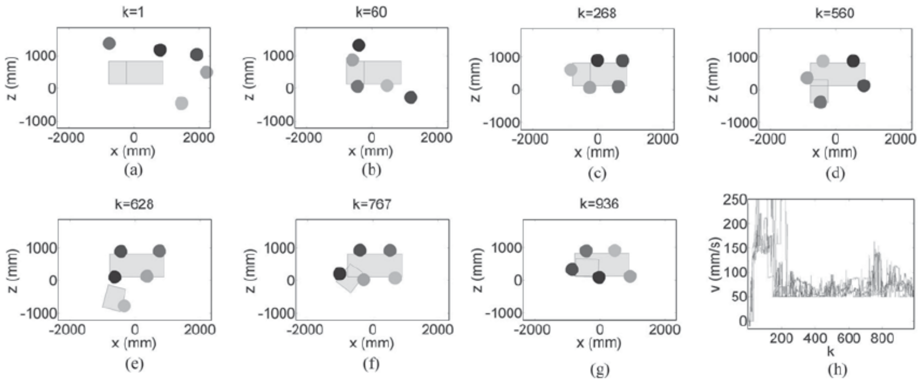


Figure 2.5: Experimental results for perimeter tracking with five robots.
(a)–(g) Snapshots of selected time points indicating the location of the spill and the agents.
(h) Linear speeds of all agents.

gence of the tracking algorithm was analyzed for multiple spills under certain conditions. Simulation results showed excellent performance in tracking multiple spills that come together or split during the process. The control algorithm proposed was then modified to fit the requirements of real experiments, and the performance was verified with a group of robotic agents. Both simulation and experiment results show that with the proposed method, the agents do successfully detect and track spills of various shapes, sizes, and movements. Figure 2.5 shows a typical experimental result for five robots with a large spill that grows, splits into two spills, and then collapses to one large spill.

A hybrid system of finite states with feedback control was developed for a group of agents with a limited sensor FOV and a limited communication range to search for, detect, and track a number of spills in an unknown environment. A hierarchical potential field was designed for agents in different states to control their movement that provides a simple control law for collision avoidance. Simulation and experiment results demonstrated that the agents do successfully detect and track spills whose location, shape, size, and number may change over time. Significantly, this adaptation occurs automatically with no need to specify a predetermined distance between adjacent agents. Further focus on search efficiency and optimal convergence rates is ongoing. Additional enhancement of the robots sensing ability continues.

Additional details of this research can be found in [3, 12, 13], reproduced in Appendix A in Section A.1.3, Section A.2.9, and Section A.2.8 respectively.

2.7 Sensor Choice Optimization and Control of Heterogeneous Agents

The goal of this novel research effort was to address the impact that variations in a pre-selectable composition of a multi-agent system (MAS) can have on its resulting capabilities and performance while utilizing a composition-tunable potential-function control scheme. This was accomplished by first defining the litany of capabilities and performance characteristics of individual agents, and the relationships between these (intra-agent).

Next, various incarnations of the MAS were hypothesized where distinctions existed through variables such as agent types, payload types and capabilities, relative quantities, motion behaviors, and most notably, potential function control tuning parameters. These were extensively represented and tested both in simulation and in hardware experimentation resulting in rich data sets.

Then, trade-off analyses were conducted to identify those factors demonstrating significant importance. These analyses provided insight to the control system and hardware design such that the core elements of the MAS itself could be orchestrated to provide the best efficiencies against comprehensive mission accomplishment.

Finally, the analyses provided insight to mission planners such that they would be able to tailor the composition of a task-specific MAS in terms of cost and performance.

A novel implementation of potential function control is developed that utilizes tuning parameters generated from state space relativity. For example, a tuning parameter consisting of the relative charge ratios of the individual agents and state space occupancy map cells is manipulated to optimize agent dispersion, path planning efficiency, and mission completion times. The results of this research show how the census of agents, including quantity and composition, ultimately impacts the overall performance of the MAS. For example, increases in quantity result in a non-linearly fading increase in performance measured by time to complete a pre-defined mission. This non-linearity means that more is not always better, especially in consideration of logistic burdens that increase with additional quantities. Figure 2.6 shows a characteristic final shared occupancy map from Webots [33] simulation.

The potential-function control architecture proved particularly robust and capable when used in conjunction with state space representation. Manipulation of the relative charges associated with MAS components and occupancy mapping resulted in a 25 percent improvement in task completion criteria. The dramatic improvement in time-to-completion versus the number of agents utilized is shown in Figure 2.7.

Finally, the concept of ”sensing opportunity” is postulated where the combined capability of a MAS as a function of its agents’ sensing systems is very much impacted by each sensor’s opportunity potential to collect information of value to the desired MAS behavior, and not necessarily the sum total raw capability of the combined sensor payloads.

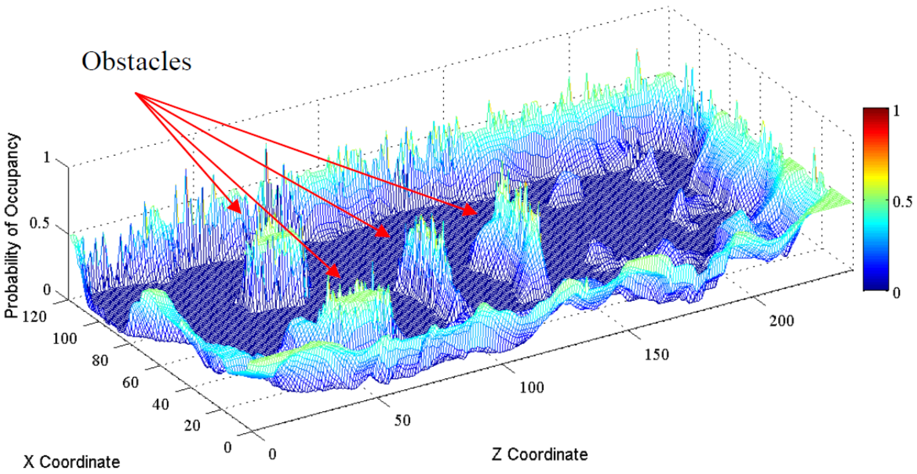


Figure 2.6: Shared map of exploration region (Webots simulation)

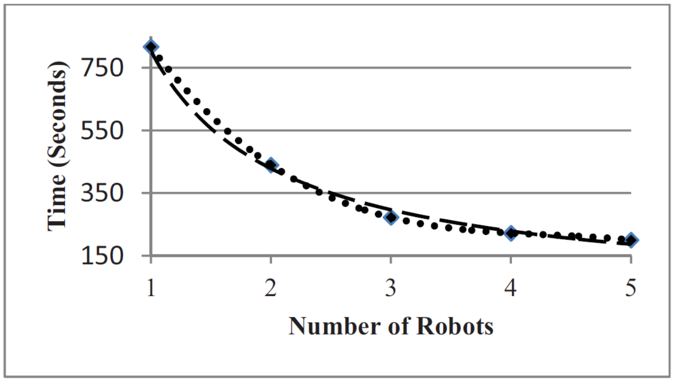


Figure 2.7: Time to complete exploration mission versus number of agents.

Full details of and additional aspects to this research can be found in [34], and partially published in [4, 15] and reproduced in Appendix A in Section A.2.11 and Section A.1.4 respectively.

2.8 Rendezvous in an Unbounded Domain without a Shared Coordinate System

The aggregation and rendezvous problems are basic hurdles in swarm robotics and distributed control. In aggregation, the goal is to develop a distributed control that allows an arbitrary number of agents to act together to achieve a common goal. For the rendezvous problem, the goal is for the agents to converge to a common position. We assume that the agents do not have a shared coordinate system and operate in an unbounded two-dimensional domain. Also, we assume communication constraints between the agents that are realistic for current technology. However, the control developed here is high level and rather abstract; we are not concerned with issues like obstacle avoidance, and the agents are assumed to be point masses operating in discrete time.

Without shared coordinates, robots must first sense the position of neighbors to develop a shared map. In this research, we explored a two stage control protocol, first utilizing random motion that allows the agents to develop shared coordinates. The agents then used local averaging to solve the rendezvous problem under some realistic constraints.

Communication restrictions between agents depend on mission specific parameters: the sensors of the robots, the size of the domain, etc. We make the following assumptions on the communication abilities of the agents and their ability to infer others' positions.

- **No shared map:** Agents must meet or sense each other in order share coordinates.

- **Sensing Radius:** Each robot is equipped with a sensor, such as a camera or laser range finder. Most commonly, such sensors have a sweep of less than 360 degrees, and exhibit sensing errors. We assume that each agent can perfectly detect all other agents within some radius S_r.

- **Simultaneous Communication:** Each agent can maintain at most k simultaneous communications. For example, standard Bluetooth technology allows $k = 7$ simultaneous communications. It's important to note that it is not necessary to keep the same k partners at each moment.

- **Probabilistic Communication Radius:** Robots have a limited energy source and may be examining an area much larger than the power of the antennae can cover. In general, the probability of a successful communication decreases with distance. We simplify matters by assuming a fixed probability p_d of the failure of each communication attempt.

We developed a control where agents first move randomly to create a cache of comrades with common coordinates. Then each agent picks some small number of individuals from its cache and moves to the average position of those individuals. We then developed a random motion algorithm that supplies the agents with common coordinates and provides bounds on the time through simulations. We discuss the global dynamics of the averaging portion of the control.

The challenge is to rendezvous a relatively small number of agents in a perhaps vast environment. We have given a two stage process where agents first find a shared coordinate systems and are able to communicate with each other. Then, each agent moves directly between two other agents in order to rendezvous at a common point. We choose only two others in order to limit dropped communications due to technology constraints.

We have considered distributed controls that are functions only of an agent's position. Real robots move in continuous time and continuous space, and we look to build simulations that utilize the same controls but keep track of each robot's voyage to distant positions. For example, a robot may choose a random point in its wandering domain and during its trip to that point encounter another robot within its sensing radius. These encounters were ignored in this work, and so the results achieved form a worst case scenario for the number of positions to explore (i.e., an upper bound) to achieve aggregation.

More specifically, suppose each agent has a timer based on its maximum speed and the diameter of its wandering domain. Once it picks a random point to move to, it sets its speed so that it arrives at the point as the timer is expiring. It then performs a sensing sweep at that point (based on another timer) and repeats the process. In this way the agents maintain a great deal of synchronization.

Additional details of this and related research can be found in [17, 18], reproduced in Appendix A in Section A.2.13 and Section A.2.14 respectively.

2.9 Unstructured Formation Control of Heterogeneous Agents for Exploration

In this research thrust, we developed a fully distributed, scalable method of controlling agents with nonholonomic constraints using a Morse potential function. This method successfully controls a swarm of differential-drive (unicycle-type) agents to stable and predictable formations whose structures are not defined *a priori*. The system achieves a stable, minimal energy state.

We consider the effect of *numerosity constraints*, as observed in birds and fish in their shoaling and flocking behavior as a mechanism of reducing complexity, in the interest of achieving fully distributed control over a swarm of any size. The application of numerosity constraints to a swarm system allows the swarm to grow without bound and with no increase in required processing capability of the individual agents. We explore this parameter as a method of minimizing processing and storage requirements while still achieving the qualitative swarm performance. Results from simulations are given for swarms ranging in size over N={6,...,100} acting under our proposed controller as applied to differential-drive (unicycle-type) robots.

In many applications that employ multiple autonomous agents, the ability of the agents to move in formation is highly desirable, even if the formation structure is not explicitly given. This is certainly true for environmental exploration and coverage applications (see, e.g., [35–38]), or escort applications [20, 39]. Stable formation control can reduce the occurrence of agents getting "lost" or separated from the group. In this research, we specifically examine the effect of "numerosity constraints" on the dynamic formation behavior of a group of differential-drive ground robots under fully distributed control.

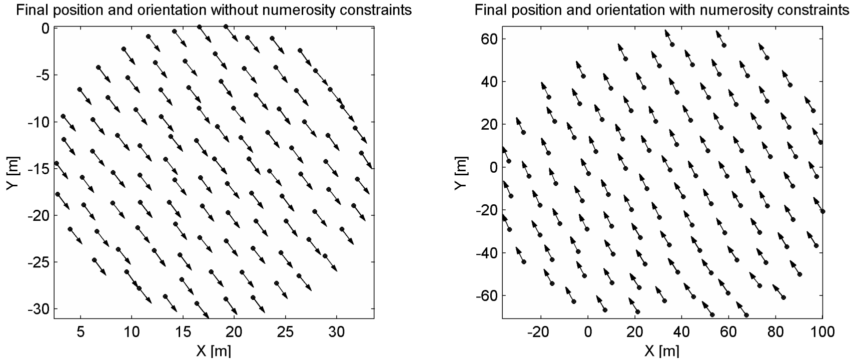
The concept of a "numerosity constraint" [40, 41] is important to consider; evidence from natural systems has shown that both birds [42] and fish [43, 44] exhibit such constraints in their shoaling and flocking behavior. Clearly, in a murmuration of starlings each individual cannot keep track of the entire flock that numbers in the thousands, nor can an individual fish keep track of the entirety of its school. Each individual is thus only aware of a finite number of its nearest neighbors, making its own motive decisions based only on this input. This ability is clearly desirable in a scalable application of multiple mobile robots, allowing each individual to bound its input requirements (and accordingly, its memory and processing requirements as well as its communication or sensing requirements) regardless of the overall size of the swarm.

The unconstrained case achieves a significantly denser structure due to the fact that each agent is aware of all other agents; thus, the presence of distant agents actually decreases the swarm spacing. In Figure 2.8a, the agents are in general separated by $3m$ to $4m$, whereas in Figure 2.8b the agents are separated by $12m$ to $13m$. In fact, for the parameters used in these simulations, the minimum-energy spacing for two agents by themselves is approximately $15.5m$. This reveals that by applying numerosity constraints, the typical spacing of agents can be regulated to "near" a design point regardless of the overall size of the swarm.

It should be noted that in these figures, the absolute positions and absolute orientations are not critical; this is the very nature of seeking such semi-stable configurations. Only the relative positions and orientations are of importance.

We successfully developed a controller that yields stable formations for agents with nonholonomic constraints. We additionally allow a constraint to be placed on the agents restricting their awareness of the bulk of the swarm in order to allow scalability, and we have shown that this constraint numerically reduces the ability of the swarm to achieve its minimum possible energy; even under this constraint, however, the swarm is still able to significantly reduce its energy relative to the starting energy, and successfully achieves formation control in a predictable way, maintaining "design parameters" for ideal spacing of agents regardless of swarm size.

Additional details of this and related research can be found in [19, 20], reproduced in Appendix A in Section A.2.15 and Section A.2.16 respectively.



(a) Steady-state formation with no numerosity constraint

(b) Steady-state formation with numerosity constraint

Figure 2.8: Steady-state formations of a swarm of 100 agents.

2.10 Conclusion

This research project attempted to develop a unified, systematic, and formal approach to address the previously identified issues in multi-agent systems. The project proposed to draw inspiration from biological systems exhibiting swarm behavior suitable for control of such interconnected agents. The basic research aimed to develop a task-oriented decentralized controller housed in each agent which derived its control action from the interactions among neighboring agents, environmental cues, and mission-at-hand. The research also aimed to develop a novel formalism to represent such complex and uncertain networks of large number of agents with the help of probabilistic graphs, and develop methodologies to analyze systems stability, robustness, and cohesiveness in the presence of connection uncertainties and estimation errors.

With the support of the Army Research Office, we have successfully expanded the field of distributed robotic control under various uncertainties, exploiting recent results in graph theory when applicable. Each of the specific topics set forth in our goals has been investigated, and all of them have been achieved to the extent possible. Our research has touched on a myriad of applications utilizing distributed control and estimation, confirming our intuition that this area of research is vital and meaningful.

We thank the Army Research Office for its support of our research. Without such support, this would not have been possible. In the course of this research work, this funding provided direct support for two graduate students, and provided indirect support (i.e., fostered collaborative research) for another graduate student, two postdoctoral scholars, and at least six undergraduate students.

Bibliography

[1] M. Kumar, D. Garg, and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," *IEEE Transactions on Automatic Control*, vol. 55, pp. 743–748, Mar. 2010.

[2] D. L. Milutinovic and D. P. Garg, "Kalman smoother based force localization and mapping using intravital video microscopy," *Journal of Dynamic Systems, Measurement, and Control*, vol. 132, no. 6, p. 061503, 2010.

[3] G. Zhang, G. K. Fricke, and D. P. Garg, "Spill detection and perimeter surveillance via distributed swarming agents," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 121–129, Feb. 2013. ONLINE: 2011 Sep.;print to appear.

[4] R. F. Young and D. P. Garg, "Modeling, simulation, and characterization of distributed multi-agent systems," *Journal of Systemics, Cybernetics and Informatics*, vol. 10, no. 2, pp. 73–80, 2012.

[5] M. Kumar, D. Garg, and V. Kumar, "Self-sorting in a swarm of heterogeneous agents," in *Proceedings of the American Control Conference (ACC2008)*, pp. 117–122, Jun. 2008.

[6] M. Kumar, D. Milutinović, and D. Garg, "Role of stochasticity in self-organization of robotic swarms," in *Proceedings of the 2008 American Control Conference (ACC2008)*, pp. 123–128, Jun. 2008.

[7] G. Zhang and D. P. Garg, "Mobile multi-robot control in target search and retrieval," in *Proceedings of the 2008 ASME Dynamic Systems and Control Conference (DSCC08)*, (Ann Arbor, MI), pp. 677–683, 2008.

[8] G. K. Fricke, D. P. Garg, and D. Milutinović, "Sensing and estimation on an experimental testbed for swarm robotics," in *Proceedings of the 2nd ASME Dynamic Systems and Controls Conference (DSCC'09)*, vol. 2, (Hollywood, CA), pp. 195–202.

[9] G. Fricke, D. Milutinović, and D. Garg, "Robotic pose estimation via an adaptive Kalman Filter using state-varying noise," in *Proceedings of the IASTED Robotics and Automation Conference (IASTED RA'09)*, (Cambridge, MA), ACTA Press, Nov. 2009.

[10] G. Fricke and D. Garg, "Discrimination and tracking of individual agents in a swarm of robots," in *Proceedings of the American Control Conference (ACC2010)*, (Baltimore, MD), pp. 2742–2747, June 30—July 2 2010.

[11] G. Fricke, A. Caccavale, and D. Garg, "Mobile sensor frame mapping via vision and laser scan matching," in *3rd International Symposium on Resilient Control Systems (ISRCS)*, (Idaho Falls, ID), pp. 43–46, IEEE, Aug. 10-12 2010.

[12] G. Zhang, D. Garg, and G. Fricke, "Hazardous spill perimeter detection and monitoring via multiple autonomous mobile robotic agents," in *Proceedings of the 3rd ASME Dynamic Systems and Control Conference (DSCC2010)*, vol. 2, (Cambridge, MA), pp. 639–646, Sept. 2010.

[13] G. Fricke, G. Zhang, W. Li, A. Caccavale, and D. Garg, "An intelligent sensing network of distributed swarming agents for perimeter detection and surveillance," in *Proceedings of the 3rd ASME Dynamic Systems and Control Conference (DSCC2010)*, (Cambridge, MA), pp. 741–748, Sep. 2010.

[14] D. Milutinović and D. P. Garg, "A samping approach to modeling and conttrol of a large size robot population," in *Proceedings of the 3rd ASME Dynamic Systems and Control Conference (DSCC2010)*, vol. 2, pp. 631–637, Sep. 2010.

[15] R. Young and D. P. Garg, "Modeling, simulation, and characterization of distributed multi-agent systems," in *Proceedings of the International Multi-Conference on Complexity, Informatics and Cybernetics*, pp. 179–184, International Institute of Informatics and Systemics, Mar. 2011.

[16] M. Kumar and D. P. Garg, "Aggregation of heterogeneous units in a swarm of robotic agents," in *Proceedings of the 4th International Symposium on Resilient Control Systems (ISRCS)*, pp. 107–112, aug. 2011.

[17] G. Fricke, B. Rogers, and D. Garg, "On the stability of swarm consensus under noisy control," in *Proceedings of the 4th ASME Dynamic Systems and Control Conference (DSCC2011)*, (Arlington, VA), pp. 291–298.

[18] B. Rogers, G. Fricke, and D. Garg, "Aggregation and rendezvous in an unbounded domain without a shared coordinate system," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, (Orlando, FL), pp. 1437–1442, Dec. 2011.

[19] G. Fricke, K. Lieberman, and D. Garg, "Swarm formations under nonholonomic and numerosity constraints," in *Proceedings of the 5th ASME Dynamic Systems and Control Conference (DSCC2012)*, (Ft. Lauderdale, FL), 17-19 Oct. 2012.

[20] K. M. Lieberman, G. K. Fricke, and D. P. Garg, "Decentralized control of multi-agent escort formation via Morse potential function," in *Proceedings*

of the 2012 ASME Dynamic Systems and Control Conference (DSCC2012), (Ft. Lauderdale, FL), Oct. 2012.

[21] T. A. Schwickert, R. L. Lindquist, G. Shakhar, G. Livshits, D. Skokos, M. H. Kosco-Vilbois, M. L. Dustin, and M. C. Nussenzweig, "In vivo imaging of germinal centres reaveals a dynamic open structure," *Nature*, pp. 83–87, 2007.

[22] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, (New York, NY, USA), pp. 25–34, ACM, 1987.

[23] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, Dec. 1998.

[24] J. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 905–908, Dec 2001.

[25] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part I: Fixed topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC2003)*, vol. 2, pp. 2010–2015, Dec. 2003.

[26] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the 40th IEEE Conference on Decision and Control, (CDC2001)*, vol. 3, pp. 2968–2973, 2001.

[27] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, pp. 401–420, March 2006.

[28] D. Milutinović, P. Lima, and M. Athans, "Biologically inspired stochastic hybrid control of multi-robot systems," in *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.

[29] D. Milutinović and P. Lima, "Modeling and optimal centralized control of a large-sized robotic population," *IEEE Transactions on Robotics*, pp. 1280–1285, 2006.

[30] D. Milutinović and P. Lima, *Cells and Robots: Modeling and Control of Large-size Agent Populations*. Springer, 2007.

[31] D. Eppstein, "Subgraph isomorphism in planar graphs and related problems," *Journal of Graph Algorithms and Applications*, vol. 3, pp. 1–27, 1999.

[32] A. D. J. Cross, R. C. Wilson, and E. R. Hancock, "Inexact graph matching using genetic search," *Pattern Recognition*, vol. 30, no. 6, pp. 953–970, 1997.

[33] Webots. http://www.cyberbotics.com.

[34] R. F. Young, *Modeling, Simulation, and Characterization of Distributed Multi-agent Systems*. PhD thesis, Duke University, 2011.

[35] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 243–255, Apr. 2004.

[36] S. Gabriele and P. Giamberardino, "Redundant coverage for noise reduction in dynamic sensor networks," *WTOS*, vol. 7, pp. 855–865, October 2008.

[37] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE Conference on Decision and Control (CDC 2008)*, pp. 3947–3952, Dec. 2008.

[38] F. Zhang, Y. Xi, Z. Lin, and W. Chen, "Constrained motion model of mobile robots and its applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 773–787, Jun. 2009.

[39] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment / escorting mission," *IEEE Robotics and Automation Magazine*, vol. 15, pp. 22–29, Mar. 2008.

[40] N. Abaid and M. Porfiri, "Collective behavior of fish shoals in one-dimensional annular domains," in *Proceedings of the American Control Conference (ACC2010)*, pp. 63–68, Jun./Jul. 2010.

[41] N. Abaid and M. Porfiri, "Consensus over numerosity-constrained random networks," *IEEE Transactions on Automatic Control*, vol. 56, pp. 649–654, Mar. 2011.

[42] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic, "Empirical investigation of starling flocks: a benchmark study in collective animal behaviour," *Animal Behaviour*, vol. 76, no. 1, pp. 201 – 215, 2008.

[43] R. W. Tegeder and J. Krause, "Density dependence and numerosity in fright stimulated aggregation behaviour of shoaling fish," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 350, no. 1334, pp. 381–390, 1995.

[44] C. Agrillo, M. Dadda, G. Serena, and A. Bisazza, "Do fish count? Spontaneous discrimination of quantity in female mosquitofish," *Animal Cognition*, vol. 11, pp. 495–503, 2008.

Appendix A

Manuscripts

This appendix includes a compilation of the manuscripts written and published in peer-reviewed journals and peer-reviewed conference proceedings.

A.1 Journal Articles

A.1.1 Segregation of Heterogeneous Units in a Swarm of Robotic Agents

The following paper (on the next 6 pages) was co-authored by Manish Kumar, Devendra P. Garg, and Vijay Kumar, and appeared in March 2010 in IEEE Transactions on Automatic Control, Volume 55, pages 743–748.

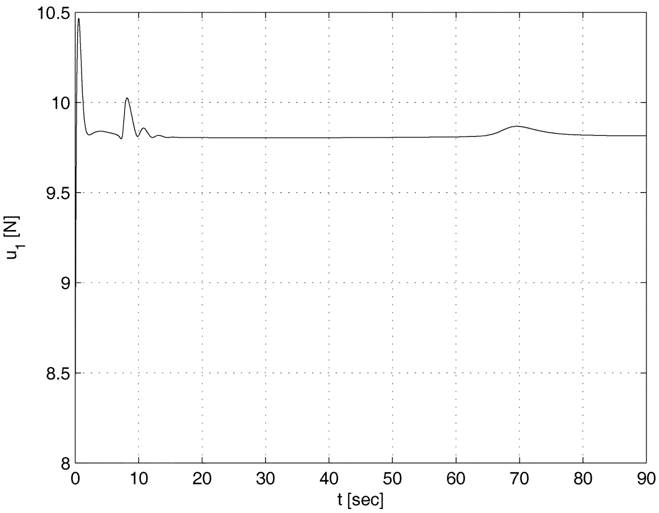


Fig. 8. Thrust force profile during landing on a sailing aircraft carrier.

force is bounded, the resulting control scheme ensures global asymptotic stability. If however both, the thrust force and the rolling moment are bounded, the proposed controllers ensure semiglobal asymptotic stability.

References

[1] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[2] J. Hauser, S. Sastry, and G. Meyer, "Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft," *Automatica*, vol. 28, no. 4, pp. 665–679, 1992.

[3] P. Martin, S. Devasia, and B. Paden, "A different look at output tracking: Control of a VTOL aircraft," *Automatica*, vol. 32, no. 1, pp. 101–107, 1995.

[4] R. Olfati-Saber, "Global configuration stabilization for the VTOL aircraft with strong input coupling," *IEEE Trans. Autom. Control*, vol. 47, no. 11, pp. 1949–1952, Nov. 2002.

[5] A. R. Teel, "Global stabilization and restricted tracking for multiple integrators with bounded controls," *Syst. Control Lett.*, vol. 18, no. 3, pp. 165–171, 1992.

[6] H. Ye, H. Wang, and H. Wang, "Stabilization of a PVTOL aircraft and an inertia wheel pendulum using saturation technique," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 6, pp. 1143–1150, Nov. 2007.

[7] A. Zavala-Rio, I. Fantoni, and R. Lozano, "Global stabilization of a PVTOL aircraft model with bounded inputs," *Int. J. Control*, vol. 76, no. 18, pp. 1833–1844, 2003.

[8] A. Hably, F. Kendoul, N. Marchand, and P. Castillo, "Further results on global stabilization of the PVTOL aircraft," in *Positive Systems*, C. Commault and N. March, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 303–310.

Segregation of Heterogeneous Units in a Swarm of Robotic Agents

Manish Kumar, Devendra P. Garg, and Vijay Kumar

***Abstract*—There are several examples in natural systems that exhibit the self-organizing behavior of segregation when different types of units interact with each other. One of the best examples is a system of biological cells of heterogeneous types that has the ability to self-organize into specific formations, form different types of organs and, ultimately, develop into a living organism. Previous research in this area has indicated that such segregations in biological cells and tissues are made possible because of the differences in adhesivity between various types of cells or tissues. Inspired by this differential adhesivity model, this technical note presents a decentralized approach utilizing differential artificial potential to achieve the segregation behavior in a swarm of heterogeneous robotic agents. The method is based on the proposition that agents experience different magnitudes of potential while interacting with agents of different types. Stability analysis of the system with the proposed approach in the Lyapunov sense is carried out in this technical note. Extensive simulations and analytical investigations suggest that the proposed method would lead a population of two types of agents to a segregated configuration.**

***Index Terms*—Differential potential, heterogeneous swarm, segregation.**

I. Introduction

Control of multiple vehicles carrying out a cooperative task has received the attention of several researchers working in the area of mobile robotics because of its potential applications in a number of fields including large-scale sensing operations, cooperated search and rescue task, surveillance, reconnaissance, and boundary protection. One of the very first applications of cooperative control of multiple agents was behavioral simulation of flocks of birds, herd of animals and schools of fish for computer graphics presented by Reynolds [11]. He noted three simple behaviors that led to flocking in birds and fish: collision avoidance, velocity matching, and flock centering (in decreasing order of precedence). Based on inspirations drawn from Reynolds' approach, many researchers have focused on designing a decentralized controller for achieving flocking behavior. The examples of these decentralized control mechanisms include techniques based on behavioral control [2], leader-follower [5], and artificial potential function [9], [10].

Obtaining a desired shape and pattern of the formation can be critical for a specific mission relying on coordinated action by multiple mobile agents. For example, if a large number of robots need to be deployed to perform complicated tasks such as surveillance of a large area, perimeter protection of a vital installation, or surrounding site of

Manuscript received October 24, 2007; revised September 20, 2008, June 30, 2009, and November 18, 2009. First published February 08, 2010; current version published March 10, 2010. Recommended by Associate Editor C. J. Tomlin.

M. Kumar is with the Department of Mechanical Engineering, University of Cincinnati, Cincinnati, OH 45220-2872 USA (e-mail: manish.kumar@uc.edu).

D. P. Garg is with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708-0001 USA (e-mail: dpgarg@duke.edu).

V. Kumar is with the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104-4206 USA (e-mail: kumar@seas.upenn.edu).

Color versions of one or more of the figures in this technical note are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2010.2040494

a chemical or hazardous waste spill, the robots must be able to autonomously organize themselves in certain formation, pattern, or shape. In many situations, it may not be possible to integrate all of the capabilities, sensing or actuation, required for different kinds of tasks in an individual robot. Accordingly, the robots may have heterogeneous abilities for sensing and actuation that will enable them to perform specific tasks. Heterogeneous robots must be able to self-organize themselves in a mission specific manner to carry out the tasks assigned to them cooperatively. This technical note focuses on one particular behavior of segregation in a swarm of heterogeneous agents.

Segregation is a phenomenon which is seen in several biological systems. For example, ants sort brood [4] in concentric annuli with the smallest members (need least tending) in the middle and the largest members (need most tending) on the periphery. Several species of cockroaches aggregate or segregate [1] based on odors of strains. Similarly, some anuran species of amphibian larvae [8] have shown to recognize and associate with their siblings or half-siblings. Segregation of cells based on their types and functionalities is one of the best examples of sorting in biological systems. Cell segregation is one of the basic phenomenon which leads to formation of patterns and organs in living organisms. Complex patterns form in living systems as a result of interactions among basic constituent units. Examples include the formation of ocular dominance stripes in the visual cortex of cats and monkeys, and the formation of the tentacle pattern in hydra.

Study of the mechanisms via which these patterns form can provide valuable insights for distributed problem solving strategies. Most of the models in literature that explain the formation of patterns rely on the principles of differential attraction/inhibition. For example, Swindale's model [14] accounts for the formation of ocular dominance stripes in the visual cortex based on the local activation and lateral inhibition (LALI) mechanism [6] for the same type of synapses, and the local inhibition and lateral activation (LILA) mechanism [3], [14], the reverse, for the unlike type of synapses. Similarly, reaction-diffusion [17] is a model to mathematically represent the transport phenomenon in biological and natural systems. This model tries to explain the interaction of particles with the environment and their motion in space. One of the models to explain the brood sorting in ants is based on *differential diffusion*. This model [12] achieves sorting in two phases. In the first phase, called clustering phase, broods of all types are clustered around the same place. In the second phase, called spacing phase, ants move broods out in a random fashion where the frequency and the distance of movement depends on the weight of the brood.

In early 1990s, Graner and Glazier [7] proposed a lattice based modified version of large-Q Potts model with differential adhesivity to explain and simulate the segregation of a mixture of two types of biological cells. In fact, it has been long known [13] that it is the difference in intercellular adhesivity that leads to sorting in cells. The final state of cell configuration is achieved when the overall surface energy is globally minimized. Based on this principle, Steinberg [13] postulated that two types of cellular units A and B are segregated when:

$$W_{AB} < \frac{(W_{AA} + W_{BB})}{2} \tag{1}$$

where W_{AA} and W_{BB} represent the work of cohesion between cells of same types (i.e., between types A & A, and B & B respectively), and W_{AB} represents the work of adhesion between cells of types A and B. The method for segregation in artificial mobile agents presented in this technical note is motivated by this differential adhesivity phenomenon observed in biological systems.

The primary contribution of this technical note is in the synthesis and analysis of a controller that allows heterogeneous robots to segregate so that they form separate groups comprising of homogeneous robots. The specific problem is formulated in the next section. This is followed by the formulation and the analysis of the control law. Finally, an extensive simulation study is presented as a means to validate the approach presented in this technical note.

II. Problem Formulation

The group of mobile agents, considered in this technical note, consists of N fully actuated agents, each of whose dynamics is given by the double integrator

$$\dot{q}_i = p_i \quad \text{and} \quad \dot{p}_i = u_i \qquad i = 1, 2, \ldots, N \tag{2}$$

where q_i and p_i are m-dimensional position and velocity vectors respectively of agent i. The group of mobile agents consists of two different types of agents: type A and type B. The number of agents of type A is N_A and that of type B is N_B such that $N = N_A + N_B$. The objective of this technical note is to synthesize a controller that can asymptotically flock and separate the robots of type A and type B into two different groups (referred to as *segregation*). Agents are said to flock (asymptotically) when all agents achieve the same velocity vector and the distances between the agents are stabilized. A group of agents of types A and B are said to be *segregated* if the average distance between the agents of the like types (type A or type B) is less than the average distance between the agents of the unlike types (i.e., between the agents of type A and type B). Or

$$r_{avg}^{AA} < r_{avg}^{AB}, \quad r_{avg}^{BB} < r_{avg}^{AB} \tag{3}$$

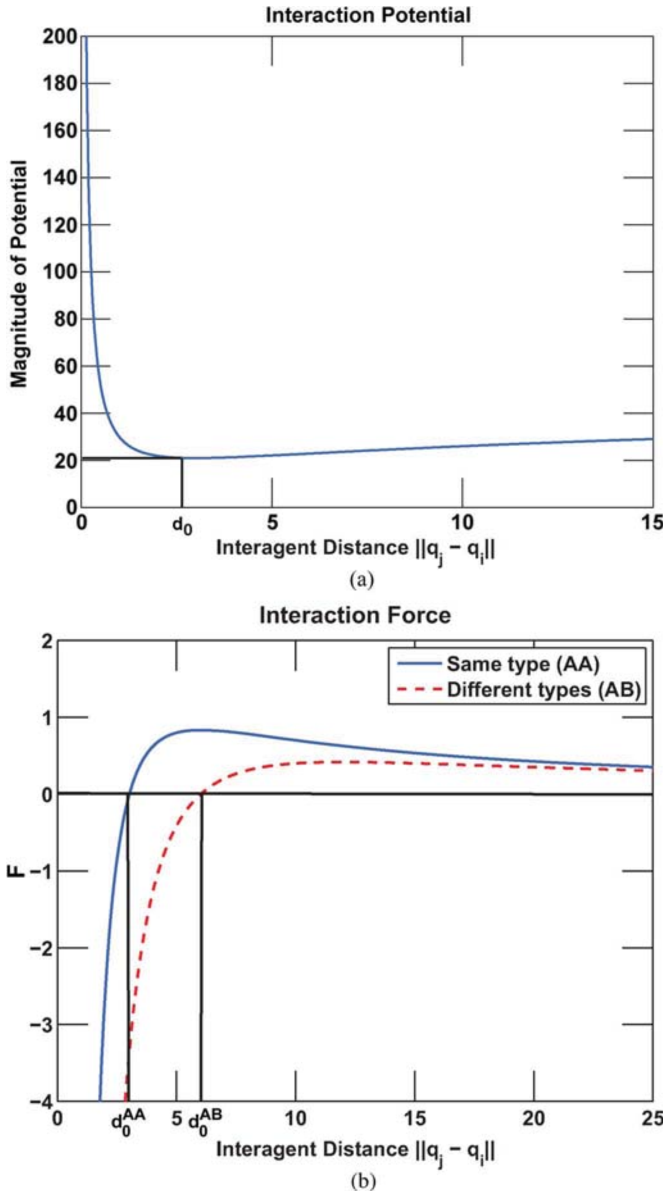
where r_{avg}^{XY} is the average distance between the agents of types X and Y.

III. Control Law Formulation

This section presents the control law which enables a population of heterogeneous agents to asymptotically flock as well as segregate. For a system of N mobile agents with N_A agents of type A and N_B agents of type B, the following feedback control law is considered:

$$u_i = -\sum_j \nabla_{q_i} V_{ij} \left(\|q_j - q_i\|\right) - \sum_j (p_i - p_j) \tag{4}$$

where u_i is the control input to the agent i, $V_{ij}(\|q_j - q_i\|)$ is the artificial potential of interaction between agents i and j, $j \in \{A, B\} \setminus i$, $\|q_j - q_i\|$ is the Euclidean norm of the vector $(q_j - q_i)$, and ∇_{q_i} is the gradient with respect to the coordinates of the agent i i.e., q_i. First term in (4) represents the gradient of potential function, and the second term



(a)

(b)

Fig. 1. Interaction potential and force. (a) Interaction potential versus inter-agent distance; (b) interaction force between agents.

represents damping and causes the agents to match their velocities with each other. The artificial potential $V_{ij}(\|q_j - q_i\|) : R^m \rightarrow R_{>0}$ is a positive function of relative distance between a pair of agents [9] given by

$$V_{ij}\left(\|q_j - q_i\|\right) = a\left(ln\left(\|q_j - q_i\|\right) + \frac{d_0}{\left(\|q_j - q_i\|\right)}\right) \qquad (5)$$

where, a is a scalar control gain, and d_0 is a parameter. Fig. 1(a) shows the potential function plotted against the inter-agent distance. As indicated in the figure, the potential becomes minimum when the inter-agent distance is d_0. The initial condition and dynamics exclude the situations where $q_i = q_j$ where the control law (4) is undefined.

Since there are two types of mobile agents involved in the system, there are three different kinds of artificial potentials involved: a) Potentials arising due to interactions between types A and A (V_{ij}^{AA}), b) Potentials arising due to interactions between types B and B (V_{ij}^{BB}), and c) Potentials arising due to interactions between types A and B (V_{ij}^{AB}). Our main result for cause of segregation is based on the concept of differential potential, i.e., agents experience different magnitudes of potential when they are interacting with the agents of different types. This is achieved by assuming different values of the parameter d_0 for different types of interaction potentials. Explicitly, in (5)

$$d_0 = \begin{cases} d_0^{AA} = d_0^{BB} & \text{if } (i, j \in A) \text{ or } (i, j \in B) \\ d_0^{AB} & \text{if } (i \in A, j \in B) \text{ or } (i \in B, j \in A). \end{cases} \qquad (6)$$

The control law for segregation can be achieved when

$$d_0^{AA} = d_0^{BB} < d_0^{AB} \qquad (7)$$

Fig. 1(b) shows the plot of force of interaction due to similar types and due to dissimilar types of robots versus inter-agent distance when the condition (7) for segregation controller is met. In this case, it can be seen that the interaction force between agents of same types is greater than that between agents of different types at any given distance. Hence, this method of segregation, based on differential potential, is analogous to Steinberg's [13] explanation of cellular segregation based on differential adhesiveness (see (1)).

IV. Controller Analysis

In this section, we carry out an analysis of convergence and stability properties of the system of multiple agents obeying dynamics given by (2) under control law given by (4). In order to carry out the stability analysis of the collective motion of the agents, the following positive definite function can be chosen as the Lyapunov function:

$$\phi(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{p} \qquad (8)$$

where $\mathbf{q} \in R^{mN}$ is stacked position vector of all agents, $\mathbf{p} \in R^{mN}$ is stacked velocity vector of all agents, and $V(\mathbf{q}) : R^{mN} \rightarrow R_{>0}$ is the total potential energy of the system which consists of three parts resulting from interactions between agents of: 1) type A, 2) types A and B, and 3) type B. This can be written as

$$\begin{aligned} V(\mathbf{q}) &= V_{AA}(\mathbf{q}) + V_{BB}(\mathbf{q}) + V_{AB}(\mathbf{q}) \\ &= \frac{1}{2}\sum_{i \in A}\sum_{j \in A, j \neq i} V_{ij}\left(\|q_j - q_i\|\right) \\ &\quad + \sum_{i \in A}\sum_{j \in B} V_{ij}\left(\|q_j - q_i\|\right) \\ &\quad + \frac{1}{2}\sum_{i \in B}\sum_{j \in B, j \neq i} V_{ij}\left(\|q_j - q_i\|\right). \end{aligned} \qquad (9)$$

The collective dynamics of the system is given by

$$\dot{\mathbf{q}} = \mathbf{p} \qquad (10)$$

$$\dot{\mathbf{p}} = -\nabla V(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{p} \qquad (11)$$

where $\hat{L}(\mathbf{q}) \in R^{mN \times mN}$ is m-dimensional graph Laplacian (see [10]). Graph Laplacian represents the interconnections present in a graph in a matrix form, and can be obtained from other graph theoretic quantities such as degree matrix and adjacency matrix. Among other important properties of graph Laplacian matrix $\hat{L}(\mathbf{q})$, it is a positive semi-definite matrix.

Lemma 4.1: Consider a system of N mobile agents. Each of the agents follows the dynamics given by (2), and with feedback control law given by (4). For any initial condition belonging to the level set of $\phi(\mathbf{q},\mathbf{p})$ given by $\Omega_C = \{(\mathbf{q},\mathbf{p}) : \phi(\mathbf{q},\mathbf{p}) \leq C)\}$ with $C > 0$, and when the underlying graph of the system is fully connected all the time, then the system asymptotically converges to the largest invariant set in $\Omega_I \subset \Omega_C$. The points in the largest invariant set in Ω_I have a velocity that is bounded, the velocities of all agents match, and the total potential of all agents given by (9) approaches a local minimum.

Proof: Differentiating $\phi(\mathbf{q},\mathbf{p})$ with respect to time and using (11) one gets

$$\begin{aligned}\dot{\phi}(\mathbf{q},\mathbf{p}) &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\dot{\mathbf{p}}\\ &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\left(-\nabla V(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{p}\right)\\ &= -\,\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p} = -\sum_i\sum_j \|p_j - p_i\|^2 \leq 0. \end{aligned} \tag{12}$$

From Lasalle's Invariance Principle, all solutions of the system starting in Ω_C will converge to the largest invariant set in $\Omega_I = \{(\mathbf{q},\mathbf{p}) \in \Omega_C : \dot{\phi}(\mathbf{q},\mathbf{p}) = 0)\}$, and this happens when the velocities of all agents match. For a detailed proof of this, please see [9], [10]. Furthermore, in the steady state, the velocities of the agents do not change, i.e., $\dot{\mathbf{p}} = 0$. For a proof of this result, see [15], [16]. From (11), it follows that at the steady state:

$$\nabla V(\mathbf{q}) = 0. \tag{13}$$

Hence, the total potential of all agents is locally minimized. ■

The gradient in (13) for an agent $i \in A$ is given by the equation

$$\begin{aligned}\nabla_{q_i}V(\mathbf{q}) = &\sum_{j\in A, j\neq i} a\left[\frac{1}{\|q_j - q_i\|} - \frac{d_0^{AA}}{\|q_j - q_i\|^2}\right]\frac{(q_j - q_i)}{\|q_j - q_i\|}\\ &+\sum_{j\in B} a\left[\frac{1}{\|q_j - q_i\|} - \frac{d_0^{AB}}{\|q_j - q_i\|^2}\right]\frac{(q_j - q_i)}{\|q_j - q_i\|} = 0. \end{aligned} \tag{14}$$

Equation (14) can be rewritten as

$$\sum_{j\in A, j\neq i} F_{ij}^{AA}(q_j - q_i) + \sum_{j\in B} F_{ij}^{AB}(q_j - q_i) = 0 \tag{15}$$

where

$$\begin{aligned}F_{ij}^{AA} &= \left[\frac{1}{\|q_j - q_i\|^2} - \frac{d_0^{AA}}{\|q_j - q_i\|^3}\right]\\ F_{ij}^{AB} &= \left[\frac{1}{\|q_j - q_i\|^2} - \frac{d_0^{AB}}{\|q_j - q_i\|^3}\right]. \end{aligned} \tag{16}$$

If we sum up (15) for all $i \in A$, and noting that $F_{ij}^{AA} = F_{ji}^{AA}$, then we will obtain the following equation:

$$\sum_{i\in A}\sum_{j\in B} F_{ij}^{AB}(q_j - q_i) = 0. \tag{17}$$

Equation (17) leads to the following proposition:

Proposition 4.2: A system of heterogeneous swarming agents consisting of two types of agents and following dynamics (2) and control law (4) flock together such that the average distance between the agents of different types (r_{avg}^{AB}) is lower bounded by the parameter d_0^{AB}, i.e., $r_{avg}^{AB} \geq d_0^{AB}$.

Proof: Since F_{ij}^{AB} is a scalar quantity, in m dimensional space, (17) can be equivalently written into the following m scalar equations:

$$\sum_{i\in A}\sum_{j\in B} F_{ij}^{AB} q_{ij}^r = 0 \qquad r = 1, 2, \ldots m \tag{18}$$

where $q_{ij}^r = (q_j^r - q_i^r)$ represents the distance along the rth dimension. Hence, $q_{ij} = \sqrt{\sum_{r=1}^m (q_{ij}^r)^2}$. Let us assume that out of $N_A N_B$ possible terms of F_{ij}^{AB}, s is the number of terms for which $F_{ij}^{AB} > 0$ and t ($s + t = N_A N_B$) is the number of terms for which $F_{ij}^{AB} \leq 0$. Hence, (18) can be written as

$$\sum_{k=1}^{s} F_k^{AB} q_k^r + \sum_{l=1}^{t} F_l^{AB} q_l^r = 0 \quad r = 1, 2, \ldots m. \tag{19}$$

Note that, for the ease of notations, the subscript ij has been replaced by subscripts k and l. If we assume: $\sum_{k=1}^{s} F_k^{AB} q_k^r = c_r$ then $\sum_{l=1}^{t} F_l^{AB} q_l^r = -c_r$.

Then, for the terms with $F_k^{AB} > 0$

$$\sum_{k=1}^{s}\sqrt{\sum_{r=1}^{m}(F_k^{AB}q_k^r)^2} \geq \sqrt{\sum_{r=1}^{m}\left(\sum_{k=1}^{s}F_k^{AB}q_k^r\right)^2}. \tag{20}$$

Noting that $q_k = \sqrt{\sum_{r=1}^m (q_k^r)^2}$, the LHS of (20) becomes

$$\sum_{k=1}^{s}\sqrt{\sum_{r=1}^{m}(F_k^{AB}q_k^r)^2} = \sum_{k=1}^{s}F_k^{AB}q_k \tag{21}$$

RHS of (20) becomes

$$\sqrt{\sum_{r=1}^{m}\left(\sum_{k=1}^{s}F_k^{AB}q_k^r\right)^2} = \sqrt{\sum_{r=1}^{m}(c_r)^2} = c. \tag{22}$$

Hence, (20) can be written as

$$\sum_{k=1}^{s}F_k^{AB}q_k \geq c. \tag{23}$$

Similarly, for the terms with $F_l^{AB} \leq 0$

$$\sum_{l=1}^{t}F_l^{AB}q_l \leq -c. \tag{24}$$

From (16), it is evident that each term of $F_k^{AB} > 0$ would tend towards zero when d_0^{AB} would be infinitely large. This means that each c_r (and hence c) would tend towards zero when d_0^{AB} is made large enough. Further, in order to prove that each term of $F_l^{AB} \leq 0$ would tend towards zero, we make use of the result that every local minima of the total potential $V(\mathbf{q})$ is an α—lattice and vice versa (See lemma 3 of [10]). This α—lattice is a spatial configuration in which inter-agent distances between all neighbors of the proximity graph (i.e., nearest neighbors) is equal to the value at which the pairwise inter-agent potential between the neareast neighbors is minimum. This means that the nearest *type B* neighbors of *type A* agents would be at distance d_0^{AB}, and hence each term of $F_l^{AB} \le 0$ will in fact tend to zero. Hence, from (23) and (24), for large values of d_0^{AB}, we can write

$$\sum_{k=1}^{s} F_k^{AB} q_k + \sum_{l=1}^{t} F_l^{AB} q_l \approx 0. \tag{25}$$

Using (16) and noting that $q_k = d_0^{AB} + x_k$ for $F_k^{AB} > 0$ and $q_l = d_0^{AB} - x_l$ for $F_l^{AB} \le 0$, where x_k and x_l are non-negative numbers, (25) can be written as

$$\sum_{k=1}^{s} \frac{x_k}{\left(d_0^{AB} + x_k\right)^2} - \sum_{l=1}^{t} \frac{x_l}{\left(d_0^{AB} - x_l\right)^2} = 0. \tag{26}$$

Since $\left(d_0^{AB} + x_k\right)^2 \ge \left(d_0^{AB} - x_l\right)^2$ for any pair of k and l, it is evident that

$$\sum_{k=1}^{s} x_k - \sum_{l=1}^{t} x_l \ge 0 \tag{27}$$

or

$$s x_c - t x_{c'} \ge 0 \tag{28}$$

where $x_c = (1/s)\sum_{k=1}^{s} x_k$ and $x_{c'} = (1/t)\sum_{l=1}^{t} x_l$. The average distance between agents of type A and type B is given by

$$\begin{aligned} r_{avg}^{AB} &= \frac{1}{s+t}\left[s\left(d_0^{AB} + x_c\right) + t\left(d_0^{AB} - x_{c'}\right)\right] \\ &= d_0^{AB} + \frac{1}{s+t}\left(s x_c - t x_{c'}\right) \ge d_0^{AB}. \end{aligned} \tag{29}$$

Hence, we can always choose d_0^{AB} to be arbitrarily large which in turn would make r_{avg}^{AB} arbitrarily large. ■

It may be noted that choosing an arbitrarily large value for d_0^{AB} does not make the average distance between the same type of agents, i.e., r_{avg}^{AA} and r_{avg}^{BB}, arbitrarily large. It is evident from (15) and (16) that the values of r_{avg}^{AA} and r_{avg}^{BB} would depend on the parameters d_0^{AA} and d_0^{BB} as well. In the proof for Proposition 4.2, we noted that each term of F^{AB} (positive or negative) would tend towards zero when d_0^{AB} would be infinitely large which means that the force on each agent due to potential from any agent of the other type would be negligible. Hence, the agents of the same type would aggregate or flock in the same manner as they would in the absence of the agents of the other type. Thus, choosing a value of d_0^{AB} to be large enough would ensure that the segregation happens according to condition (3).

V. Simulation Results and Discussions

Extensive simulations were carried out to verify the results obtained in the previous sections. In the simulations, the following parameters were assumed: $d_0^{AA} = d_0^{BB} = 3$, and $d_0^{AB} = 6$. Fig. 2(a) shows the configuration of a population of 20 agents (10 each of types A and B) in a 2-D space at different times during the simulation. The agents started off at a random configuration, and the control law given by (4) based on differential potential was applied to the agents. The final configuration at time $\mathrm{T} = 500$ sec shows that the agents of types A and B form two separate groups. Fig. 2(b) shows the plot of the average distances between the agents of types A and A (r_{avg}^{AA}), B and B (r_{avg}^{BB}), and A and B (r_{avg}^{AB}) versus time for the above simulation. At the final configuration, the average distances r_{avg}^{AA}, r_{avg}^{BB}, and r_{avg}^{AB} were found out to be 3.21, 3.22, and 6.97 respectively, which clearly shows that the population was segregated based on the condition given by expression (3).

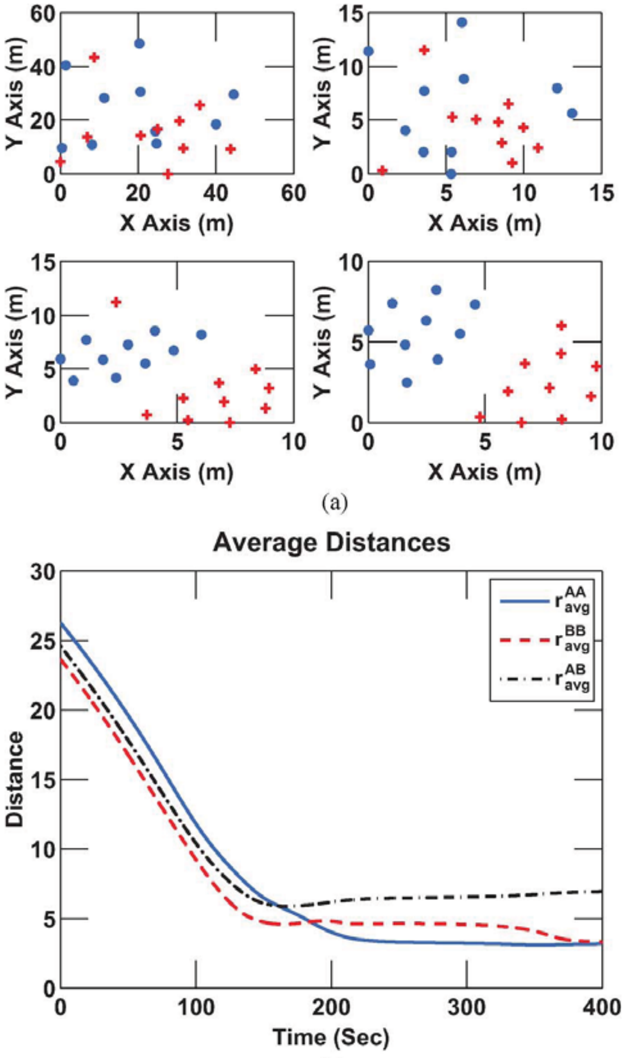


Fig. 2. Simulation with 20 agents. (a) Configurations of agents at times $\mathbf{T = 0}$ **sec** (top left), $\mathbf{T = 167}$ **sec** (top right), $\mathbf{T = 333}$ **sec** (bottom left), and $\mathbf{T = 500}$ **sec** (bottom right); (b) average distances between agents of types A and A $(\boldsymbol{r}_{avg}^{AA})$, B and B $(\boldsymbol{r}_{avg}^{BB})$, and A and B $(\boldsymbol{r}_{avg}^{AB})$.

The results given above were for only one simulation run. In order to verify that the method presented in this technical note leads to segregation in general in a population of heterogeneous agents, an extensive simulation study was carried out in which more than 100 runs were performed. Fig. 3 shows the average distance between agents at the steady-state. In each of the simulation runs, the population of agents consisting of types A and B was initialized in a random configuration obtained via uniform distribution of agents in a 2-D space, and the numbers of agents of types A and B were each chosen randomly between the values of 5 and 15. Each of the runs was carried out for 500 seconds

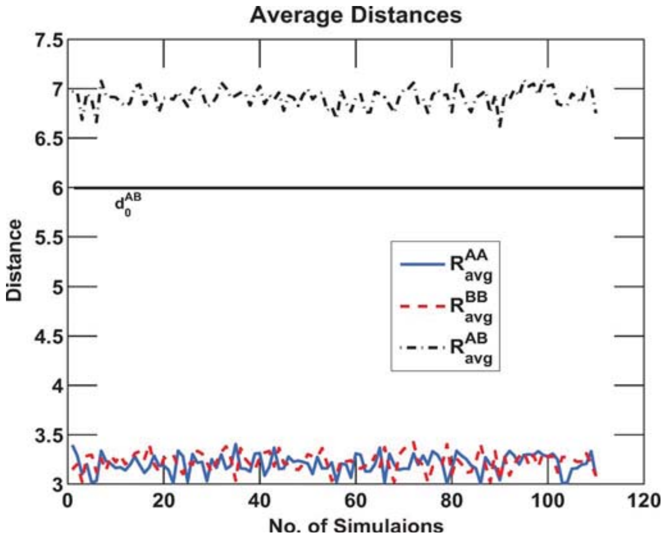


Fig. 3. Average distance between agents.

of simulation time. The average distances between the agents shown in the figure were calculated at the steady (final) state. It can be easily seen that the average distance between agents of type A (r_{avg}^{AA}) and average distance between agents of type B (r_{avg}^{BB}) is less than the average distance between agents of type A and B (r_{avg}^{AB}) for each of the simulation runs. Also, it is evident from the figure that r_{avg}^{AB} is always greater than the parameter d_0^{AB} supporting our result from Proposition 4.2.

VI. Conclusion

The technical note presents a decentralized technique to achieve a self-organized behavior of segregation in a population consisting of two different types of robotic agents. Inspired from the mechanisms by which segregation is achieved in some of the biological systems, the method presented in this technical note is based on the concept of differential artificial potential. In this differential artificial potential function framework, agents experience different magnitudes of potential when they are interacting with the agents of different types. The technical note presents the stability analysis of a population of agents in a Lyapunov framework, and lays down an analytical foundation for synthesis of controllers for segregation in artificial potential function formulation. Extensive simulation studies verify the results obtained in this technical note, and demonstrate the effectiveness of the proposed method in achieving segregation in a swarm of heterogeneous agents.

References

[1] J. Ame, C. Rivault, and J. Deneubourg, "Cockroach aggregation based on strain odour recognition," *Animal Behav.*, vol. 68, pp. 793–801, 2004.

[2] T. Balch and R. Arkin, "Behavior-based formation control for multirobot systems," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Jun. 1998.

[3] E. Bonabeau, "From classical models of morphogenesis to agent-based models of pattern formation," *Artificial Life*, vol. 3, pp. 191–211, 1997.

[4] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chreitien, "The dynamics of collective sorting: Robot-like ants and ant-like robots," in *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, J.-A. Meyer and S. W. Wilson, Eds. Cambridge, MA: MIT Press, 1991, pp. 353–363.

[5] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, Jun. 2001.

[6] A. Gierer and H. Meinhardt, "A theory of biological pattern formation," *Kybernetik*, vol. 12, pp. 30–39, 1972.

[7] F. Graner and J. A. Glazier, "Simulation of biological cell sorting using a two-dimensional extended potts model," *Phys. Rev. Lett.*, vol. 69, no. 13, pp. 2013–2016, Sep. 1992.

[8] M. A. Halverson, D. K. Skelly, and A. Caccone, "Kin distribution of amphibian larvae in the wild," *Molecular Ecology*, vol. 15, pp. 1139–1145, 2006.

[9] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Int. Conf. Decision Control*, 2001, pp. 2968–2973.

[10] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[11] C. Reynolds, "Flocks, birds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, pp. 25–34, 1987.

[12] A. B. Sendova-Franks, S. R. Scholes, N. R. Franks, and C. Melhuis, "Brood sorting by ants: Two phases and differential diffusion," *Animal Beh.*, vol. 68, pp. 1095–1106, 2004.

[13] M. S. Steinberg, "Reconstruction of tissues by dissociated cells," *Science*, vol. 141, pp. 401–411, 1963.

[14] N. V. Swindale, "A model for the formation of ocular dominance stripes," *Philosophical Trans. Royal Soc. London B*, vol. 208, pp. 243–264, 1980.

[15] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, part i: Fixed topology," in *Proc. IEEE Int. Conf. Decision Control*, 2003, pp. 2010–2015.

[16] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, May 2007.

[17] A. M. Turing, "The chemical basis of morphogenesis," *Philosophical Trans. Royal Soc. London B*, vol. 237, no. 641, pp. 37–72, 1952.

A.1.2 Kalman Smoother Based Force Localization and Mapping Using Intravital Video Microscopy

The following paper (on the next 8 pages) was co-authored by Dejan Lj. Milutinović and Devendra P. Garg, and appeared in November 2010 in Volume 132, Issue 6 of the Journal of Dynamic Systems, Measurement, and Control, pages 061503-1 – 061503-8.

Dejan Lj. Milutinović[1]
Assistant Professor
Department of Applied Mathematics and Statistics,
Baskin School of Engineering,
University of California,
Santa Cruz, CA 95064
e-mail: dejan@soe.ucsc.edu

Devendra P. Garg
Professor
Fellow ASME
Department of Mechanical Engineering and Materials Science,
Duke University,
Durham, NC 27707
e-mail: dpgarg@duke.edu

Kalman Smoother Based Force Localization and Mapping Using Intravital Video Microscopy

Motility is an important property of immune system cells. It provides cells with the ability to perform their function not only at the right time but also at the right place. In this paper, we introduce the problem of modeling and estimating an effective force field directing cell movement by the analysis of intravital video microscopy. A computational approach is proposed for solving this problem without dealing with a parametrized spatial model of the field in order to avoid potential errors due to inaccurate spatial model assumptions. We consider the dynamics of cells similar to the dynamics of distributed agents typically used in the field of swarm robotics. The method utilizes a fixed-interval Kalman filter based smoother. Its application results in a map giving the intensity and direction of the effective force field. The results show that real-time video images are a source of data, enabling us to visualize intriguing spatiotemporal phenomena inside immune system organs. The proposed approach can fill the existing gap between contemporary technology and quantitative data analyses present in the field of biosystems.

[DOI: 10.1115/1.4002485]

1 Introduction

The lymph nodes are secondary lymphoid tissues and anatomical sites, where, along with the spleen, the major component processes of the immune response to infection and vaccination occur, including antigen recognition, cellular activation, differentiation, affinity maturation, and the establishment of immune memory. Recent advances in imaging technology provide us a novel insight into the dynamics of immune cell interactions in intact lymph nodes [1–3].

In this paper, we introduce the problem of estimating the effective force field influencing cell motility based on intravital video microscopy. We name this problem *force localization and mapping* (FLAM). Considering an individual cell as an agent [4], we find this problem similar to a well-studied problem in robotics, the so-called *simultaneous localization and mapping* (SLAM) [5,6]. In both problems, the estimation is based on the individual agent's motility model and the outcome is a map that visualizes the spatial structure within the agent's environment.

In the immunological context, effective forces originate from the mechanical forces of extracellular environment and the active responses of cells to gradients of chemotactic and haptotactic substances. *Currently, there is no alternative microscopy method available for measuring these forces under physiological condition.* Therefore, we can only rely on a computational method for the effective force field estimation from intravital video microscopy. The estimation provides quantitative information that is important for understanding the behavior of the immune system as a whole. Cellular aggregation and lymphoid tissue morphology are determined largely by these fields and are of great consequence in the development of an effective immune response.

Our emphasis in solving the FLAM problem is the development of an approach that is independent of any assumption about the force field shape. In this way, we hope to avoid an error in the field estimation resulting from possibly wrong intuition, trying, for example, to match the shape of the field to visual patterns in a video sequence. By proposing this approach, however, we do not exclude the possibility that after multiple field estimations, as in the case of repeated experiments, the visual patterns can be considered. This will likely prove fruitful once confidence in the relation between the force field shape and the visual patterns has been established.

The data and model that we use in our estimation approach are described in Secs. 2 and 3, respectively. Section 4 explains the effective force estimation based on the fixed-interval optimal smoother. We discuss the limitations of this approach in Sec. 5. Section 6 describes the force estimation in the case of a time-stationary uniform field. The estimation of a nonuniform time-stationary field is introduced in Sec. 7. Finally, Sec. 8 presents the conclusions based on this research.

2 Cell Motility Data

In the case of the lymph node, cells are 50–250 μm deep in the tissue. The cell motility is observed by two-photon microscopy, which minimizes the damage to cells and surrounding tissues. Two-photon microscopy requires that the observed cells be labeled with fluorescent dyes emitting the light of a specific wavelength when excited by a laser beam. In the case of the experimental setup for lymph-node imaging that we consider, the laser beam typically sweeps out a two-dimensional region of $200 \times 150\ \mu\text{m}^2$ while its depth in the tissue is varied, providing an image stack of 50 μm thick 3D volume within the lymph node. We refer to this volume as the *visual field*.

Figure 1 shows a typical 2D image projection of the intra-lymph node image stack in which T-cells and dendritic cells are labeled with two different colors [4]. The sampling interval between subsequent scans of the visual field is in the range from 18s to 21s. The data we derive from the stacks are the cell position samples. If each individual cell, i.e., its trajectory, is uniquely labeled by the integer number i, the data we are dealing with are coordinates of cell center x_k^i, y_k^i, and z_k^i corresponding to the time points t_k^i along the ith trajectory.

Measurement of cell positions from these video images is not error free. To minimize errors, it is necessary to process 3D image stacks and to compensate for distortion introduced by the optical system and transmission of the light through the tissue prior to measuring the cell positions. Errors in the cell position measure-

[1] Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received October 3, 2008; final manuscript received April 10, 2010; published online October 29, 2010. Assoc. Editor: Loucas Louca.

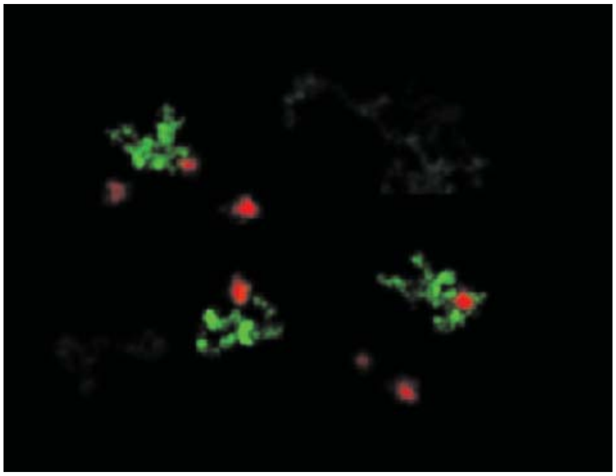


Fig. 1 Computer-generated image of a cellular interaction inside the lymph node: T-cells (red), and dendritic cells (green) [4]

ment are mainly to be attributed to finite image resolution and the significant and random change in cell morphology. Moreover, each cell must be assigned to a single trajectory, a task that may be nontrivial when two cell trajectories are in close proximity.

An example of a 2D projection of trajectories is depicted in Fig. 2 in which each trajectory is translated back to the origin (0,0). This type of diagram has been previously used [1] to detect the presence of the force. It is obvious that, while we can use the help of such a diagram to detect a constant force, it is not useful if the force field is nonuniform over the visual field. For example, in the case of data presented in Fig. 2, we can see that the force component along the x axis is nonzero and may be constant but it is difficult to say anything about y-dependence of the force field based on a figure of this type. The cell trajectories presented in Fig. 2 are generated from a realistic model of cell motility and geometry of the visual field.

We focus on the analysis of data from B- and T-lymphocytes, which have more or less rounded shapes. However, the same analysis can be also applied to motility of irregularly shaped cells such as macrophages, neutrophils, and dendritic cells. It is only necessary that the time-space scale of interest allows that the cell trajectories are fairly well described by the cell center positions. Uncertainty of the center position due to the cell shape irregularity can be assigned to the intensity of measurement errors.

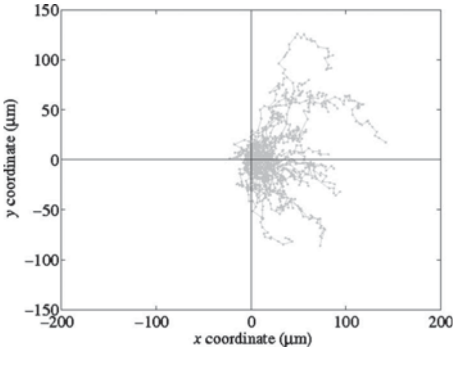


Fig. 2 Two-dimensional projection of 40 cell tracks from the 200×150×50 μm^3 visual field provided by a two-photon video microscopy (model generated). Each track is translated so that the track begins at the diagram origin. The cells can move in all directions, therefore, the limits for plotting translated trajectories are ±200 μm and ±150 μm along x and y axes, respectively.

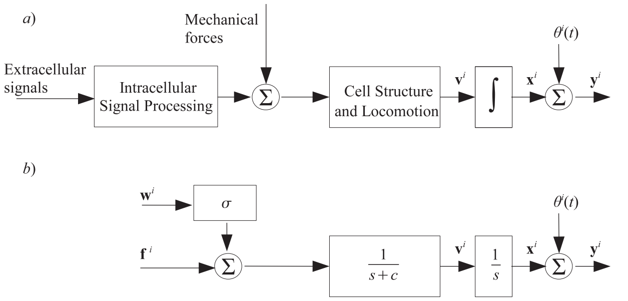


Fig. 3 Time-continuous cell motility model: (*a*) general model and (*b*) stochastic motility model

3 Cell Motility Model

According to our general model presented in Fig. 3(*a*), an individual cell senses the environment using its receptors and processes received signals intracellularly. Intracellularly generated forces that act on the cell structure arise as a result of this processing. These forces along with mechanical forces act on the cell structure, finally resulting in cell movement. While there are attempts to understand chemotaxis in more detail [7–9], in order to estimate effective forces influencing cell motility, we use a general model without going into details of intracellular signaling and cell structural properties. This double integrator model representing particle dynamics [10–12] is common in representing the motion of agents in a multi-agent systems.

The model describes motility of an individual ith cell (agent) volume center. The integrator block of the model (see Fig. 3(*a*)) diagram does not describe the property of the cell but the natural relation between cell velocity $\mathbf{v}^i$ and its position $\mathbf{x}^i$. The same relation is presented by the Laplace domain transfer function $1/s$ of the model in Fig. 3(*b*).

Neither the velocity $\mathbf{v}^i$ nor the cell position $\mathbf{x}^i$ can be measured directly. The data we measure directly are cell positions $\mathbf{y}^i$ corrupted by measurement errors θ^i. The error sources are due to limited image resolution, light diffusion, and other unpredictable disturbances influencing the detection of the cell volume center. In the model, we assume that the measurement errors $\theta^i(t)$ are additive (see Fig. 3), i.e.,

$$\mathbf{y}^i(t) = \mathbf{x}^i(t) + \theta^i(t) \tag{1}$$

In this notation, vectors $\mathbf{x}^i, \mathbf{v}^i \in R^D$, where $D=3$ is the spatial dimension. The last equation is called the *observation model* and the additive influence of errors $\theta(t)$ is depicted by the summation symbol Σ in Fig. 3(*a*), as well as in the model in Fig. 3(*b*). We assume that the error distribution is Gaussian zero mean, $\theta^i(t) \sim N(0,\Theta)$, and covariance Θ.

In general, the cell motility velocity is a result of mechanical forces of extracellular environment and the active responses of cells to gradients of chemotactic and haptotactic substances, i.e., extracellular signals (see Fig. 3(*a*)). In the model we deal with in this paper (Fig. 3(*b*)), a persistent component of these forces influencing the ith cell is described by the effective force vector $\mathbf{f}^i \in R^D$. The random force $\mathbf{w}^i$ scaled with σ accounts for the stochastic nature of sensed extracellular signals and complex mechanical forces influencing cell motility. We model this force as a D dimensional Gaussian white noise process of unit intensity $\mathbf{W} = \mathbf{I}_{D\times D}$ and uncorrelated with measurement errors.

In our model, the relation between the force and velocity is described by the transfer function with the single real pole $-c$, $c \geq 0$. The reason for this lies in the fact that available data presenting the traveled cell distance versus the square root of time $\sqrt{t}$ are bent for the small values of the time, and this is what this simple model predicts (see Fig. 4 and Appendix A).

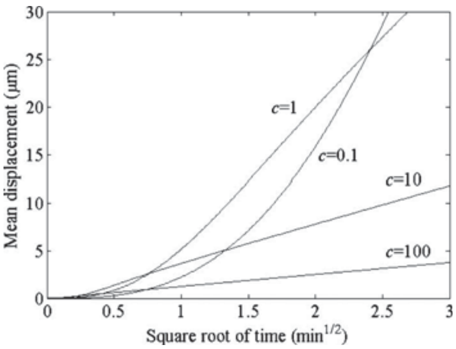


Fig. 4 Cell displacements versus the square root of traveled time: parameters $\sigma=10$ and $c=0.1$, 1, 10, and 100

Only within the limit of large $c\to\infty$ does the curvature of the diagram vanish, which means that even the highest frequency components of the force propagate to the velocity. This results in the Brownian model of cell motility, which seems unrealistic because it is expected that the mechanical structure of a cell has low-pass frequency characteristics.

In the time domain, our model of the cell i motility is given by the stochastic differential equation with the force term

$$\dot{\mathbf{x}}^i(t) = \mathbf{v}^i(t) \tag{2}$$

$$\dot{\mathbf{v}}^i(t) = -c\mathbf{v}^i(t) + \sigma\mathbf{w}^i(t) + \mathbf{f}^i(t) \tag{3}$$

We assume that the force $\mathbf{f}^i(t)$ results from a time-stationary force field, which means that the force sensed by the cell depends only on its position. Consequently, we can write $\mathbf{f}^i(t)=\mathbf{F}(\mathbf{x}^i(t))$, where function $\mathbf{F}(\mathbf{x})$ maps the position $\mathbf{x}$ to the force vector and defines the time-stationary force field that we estimate in this work. In the multi-agent literature [12], the right-hand side of Eq. (3) is a control of the agent motility, i.e., cell motility. The structure of control and stability analysis has been discussed in Ref. [12], and the influence of stochasticity term has been discussed in Ref. [13].

Let us define the column vector $\mathbf{X}^i=[\mathbf{x}^i;\mathbf{v}^i]_{2D\times 1}$, the so-called state vector. Now, we can write the motility and observation model in the matrix form as

$$\dot{\mathbf{X}}^i(t) = \mathbf{A_c}\mathbf{X}^i(t) + \mathbf{L_c}\mathbf{w}^i(t) + \mathbf{B_c}\mathbf{f}^i(t)$$

$$\mathbf{y}^i(t) = \mathbf{C_c}\mathbf{X}^i(t) + \theta^i(t) \tag{4}$$

in which matrices $\mathbf{A_c}$, $\mathbf{L_c}$, and $\mathbf{C_c}$, as well as the corresponding noise intensity matrix $\mathbf{W}$, can be easily identified (see Appendix B). Since we are dealing with sampled data, it is useful to introduce a discrete-time model,

$$\mathbf{X}^i(k+1) = \mathbf{A_d}\mathbf{X}^i(k) + \mathbf{L_d}\mathbf{w}^i(k) + \mathbf{B_d}\mathbf{f}^i(k)$$

$$\mathbf{y}^i(k) = \mathbf{C_d}\mathbf{X}^i(k) + \theta^i(k) \tag{5}$$

The definitions of the corresponding matrices $\mathbf{A_d}$ and $\mathbf{L_d}$ are provided in Appendix B. The variable k corresponds to the time point t_k at which the position vector $\mathbf{y}^i$ is sampled along the ith cell trajectory. We do not provide here the expression for the matrix $\mathbf{B_d}$ because it depends on properties of the continuous signal $\mathbf{f}^i(t)$ [14]. Moreover, in the next section, we will incorporate the signal $\mathbf{f}^i(t)$ into the state vector.

The distance and time units that we use from here on are micrometers and minutes. In this paper, we assume that the parameters σ and c of the model are equal for the same cell type. Based on available data, we take these parameters to have the values $\sigma=10$ and $c=1$. The intensity of the measurement error Θ is a diagonal matrix with diagonal elements equal to 2.5^2, corresponding to approximately one-quarter of the expected cell diameter in the case of T- or B-lymphocytes.

4 Force Model and Force Estimation

The force field $\mathbf{F}$ that we estimate is stationary and depends only on the position $\mathbf{x}$. One way to estimate the force field is to assume an analytical shape of the field, parametrize the shape, and estimate its parameters. The drawback of this approach is that the assumed form of the field can be unrealistic. In the approach that we are presenting here, we avoid any kind of structural assumptions about the function $\mathbf{F}(\mathbf{x})$ and we exploit the information about the force field from the cell trajectory data.

Under the assumption that the force field $\mathbf{F}(\mathbf{x})$ is stationary in time, the force as sensed by the ith cell is $\mathbf{f}^i(t)=\mathbf{F}(\mathbf{x}^i(t))$. Our approach to estimate $\mathbf{F}(\mathbf{x})$ is to estimate first the forces influencing cell motility along each individual cell trajectory, then use these estimations to integrate a map that visualizes the force field, which is explained in Sec. 7.

Our knowledge about the force $\mathbf{f}^i(t)$ is minimal and trajectory dependent. If the field is smooth and the cell motility is a stochastic process, the signal $\mathbf{f}^i(t)$ will be stochastic too. In the other limit, if the stochastic motility component is of small intensity, then $\mathbf{f}^i(t)$ will be smooth and possibly well described by a polynomial approximation. To include both possible limits, we model $\mathbf{f}^i(t)$ as the random process,

$$\dot{\mathbf{f}}^i(t) = \sigma_F\mathbf{w}^i_{\mathbf{F}}(t) \tag{6}$$

The parameter σ_F scales the intensity of Gaussian white noise vector $\mathbf{w}^i_{\mathbf{F}}(t)$, which has the dimension D and is uncorrelated with either $\mathbf{w}^i(t)$ or $\boldsymbol{\theta}^i(t)$. Samples from this stochastic model, although with a small probability of realization, also correspond to the variety of deterministic functions, including those with smooth force variations. The intensity of the expected variations of the force is included using the parameter σ_F. The force with faster expected variations is modeled using a larger σ_F. The force model (6) can be included into model (5), and it results in the difference equation

$$\widetilde{\mathbf{X}}^i(k+1) = \mathbf{A}\widetilde{\mathbf{X}}^i(k) + \mathbf{L}\widetilde{\mathbf{w}}^i(k)$$

$$\mathbf{y}^i(k) = \mathbf{C}\widetilde{\mathbf{X}}^i(k) + \widetilde{\theta}^i(k) \tag{7}$$

in which the force $\mathbf{f}^i(k)$ is incorporated into the augmented state vector $\widetilde{\mathbf{X}}^i=[\mathbf{X}^i;\mathbf{f}^i]_{3D\times 1}$ of the dimension $3D\times 1$ (see Appendix B).

The optimal Bayesian estimation of the state vector of the difference model such as Eq. (7) is a well-studied problem [15]. In this problem, the samples $k=1,\ k=2,\ldots,\ k=N$ are all available at the time point at which the state estimation is computed. This is the so-called fixed-interval smoothing problem.

To avoid accumulation of numerical errors and decrease the amount of matrix inversions, the realization of the used optimal smoother is in the so-called Rauch–Tung–Striebel (RTS) form of the fixed-interval optimal smoother [15],

$$\hat{\mathbf{X}}^i(k) = \hat{\mathbf{X}}^i_{\mathbf{F}}(k) + \mathbf{K_k}[\hat{\mathbf{X}}^i(k+1) - \hat{\mathbf{X}}^i_{\mathbf{F}}(k+1)] \tag{8}$$

$$\mathbf{K_k} = \mathbf{P}^i_{\mathbf{F}}(k)\mathbf{A}^T\mathbf{P}^{i-}_{\mathbf{F}}(k+1),\ \ \hat{\mathbf{X}}^i(k) = \hat{\mathbf{X}}^i_{\mathbf{F}}(k) \tag{9}$$

In our case, the computed state estimations $\hat{\mathbf{X}}^i(k)=[\hat{\mathbf{x}}^i(k);\hat{\mathbf{v}}^i(k);\hat{\mathbf{f}}^i(k)]$ include the estimation of positions $\hat{\mathbf{x}}^i(k)$, velocities $\hat{\mathbf{v}}^i(k)$, and forces $\hat{\mathbf{f}}^i(k)$ for all $k=1,\ldots,N$. The covariance matrix of the state vector estimation is

$$\mathbf{P}^i(k) = \mathbf{P}^i_{\mathbf{F}}(k) + \mathbf{K_k}[\mathbf{P}^i(k+1) - \mathbf{P}^{i-}_{\mathbf{F}}(k+1)]\mathbf{K}^{\mathbf{T}}_{\mathbf{k}} \tag{10}$$

with $\mathbf{P}^i(N)=\mathbf{P}^i_{\mathbf{F}}(N)$. The presented RTS formulation of the smoother includes the forward iteration, which implements the Kalman filter for computing vectors $\hat{\mathbf{X}}_{\mathbf{F}}$ and $\hat{\mathbf{X}}^{i-}_{\mathbf{F}}$ and matrices $\mathbf{P}^i_{\mathbf{F}}$, $\mathbf{P}^{i-}_{\mathbf{F}}$, as well as the backward iteration presented above. For details on the Kalman filter, see Appendix C.

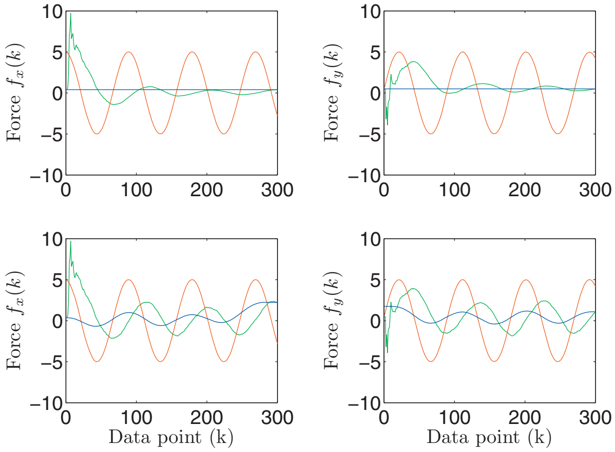


Fig. 5 The force estimation σ_F=0 (top) and σ_F=0.1 (bottom): true force (red), the forward (Kalman filter) iteration (green), and the backward iteration (blue). The true force is defined by expression (11).

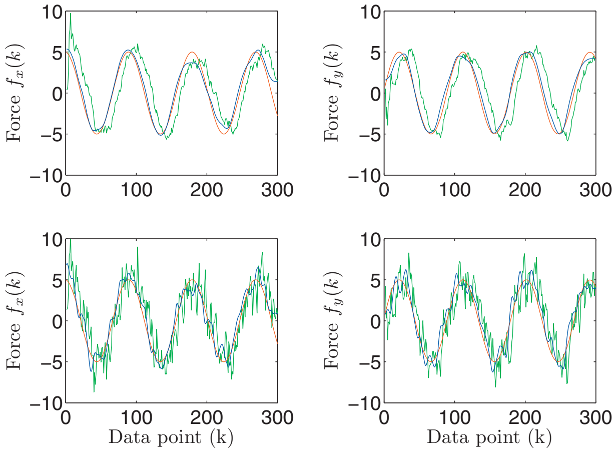


Fig. 6 The force estimation σ_F=1 (top) and σ_F=5 (bottom): true force (red), the forward (Kalman filter) iteration (green), and the backward iteration (blue). The true force is defined by expression (11).

5 Force Estimation Along the Trajectory

Among all the parameters included in the optimal smoother, the parameter σ_F can be considered as the tuning parameter. The following examples illustrate how this parameter influences the force estimation and that the stochastic model is general enough even in the case when the $\mathbf{f}^i(t)$ is smooth and deterministic.

Here, we generate the cell trajectory based on the model (Eq. (5)) and the force, which is

$$\mathbf{f}(k) = \begin{bmatrix} f_x(k) \\ f_y(k) \end{bmatrix} = \begin{bmatrix} 5\cos(4k\pi/180) \\ 5\sin(4k\pi/180) \end{bmatrix} \tag{11}$$

We are dealing here with a single cell and the index i is omitted from the notation. To make it simple, we use here the two-dimensional motility model D=2, with parameters σ=1, c=1, $\mathbf{W}$=diag$\{1,1\}$, and $\mathbf{\Theta}$=diag$\{2.5^2,2.5^2\}$.

To apply the smoother, we need to take an initial guess for the state vector estimation $\hat{\mathbf{X}}(0)$; thus, we use

$$\mathbf{x}(0) = \mathbf{y}(0) \quad \text{and} \quad \mathbf{v}(0) = \frac{1}{\mathrm{t}_1 - \mathrm{t}_0}(\mathbf{y}(1) - \mathbf{y}(0)) \tag{12}$$

because the observed values $\mathbf{y}(0)$ should be close to the true cell position. For the components of $\hat{\mathbf{X}}(0)$, relating to the force $\mathbf{f}(0)$, we use the zero vector. Following the same reasoning, we make use of the following covariance matrix of the initial state:

$$\mathbf{P}(0) = \begin{bmatrix} \mathbf{\Theta}_{2\times 2} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \dfrac{2\mathbf{\Theta}}{(t_1 - t_0)^2} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & 10^2\mathbf{I}_{2\times 2} \end{bmatrix} \tag{13}$$

In the initial state guess $\hat{\mathbf{X}}(0)$, the only arbitrary value is the zero force vector. Therefore, in the covariance vector $\mathbf{P}(0)$, we assume that the standard deviation of the force is large and we take value 10.

First, we consider the case when σ_F=0. The result of the smoothing algorithm is presented in the top panels of Fig. 5. We can see that, in the forward iteration, the force estimation changes with more data points included. When the last data point is included, the estimated force is the best estimate based on the assumption σ_F=0. Therefore, it is obvious why, in this case, the backward iteration does not improve the force estimation. The smoother estimate of the force, for all k, is identical to the last estimate of the Kalman filter in the forward iteration, which is the well-known result of the estimation theory [15].

In the bottom panels of Fig. 5, we show the result when the parameter σ_F is slightly increased. Now, σ_F=0.1 and we can see that the estimations of the forward and backward smoother iterations are different. In this case, the improvement gained by the backward iteration is considerable if we compare the phase shift between the estimated force and the true force time signal.

By increasing the parameter σ_F=1, we get a better match between the true force and the estimation in the phase and amplitude (see Fig. 6, top). In this case, σ_F is large enough so that the smoother can follow the speed of the force variations. By increasing σ_F, the smoother is able to follow even faster force variations. However, not only does a larger value for σ_F extend the dynamical range for the forces that the smoother can estimate, but it also results in a smaller attenuation of the noise propagation to the force estimation. This is illustrated in the bottom panels of Fig. 6, resulting from the smoother with σ_F=5. We can notice that the force estimation still matches the amplitude and phase of the true signal but the estimation is noisy.

These four numerical examples primarily illustrate that the force estimation along the trajectory and based on stochastic force models is feasible. They also show that the value of parameter σ_F should be small enough to suppress the propagation of the noise to the estimated force but still large enough to follow the force variation. When the parameter value is large enough, it influences the noise of estimates and not their expected values. Therefore, we see this parameter as a smoothing parameter of the force field visualization method that will be explained in Sec. 7.

6 Uniform Field Estimation

The relation between estimation performance and the cell motility model parameters and dimensions of the visual field, as well as the direction and intensity of the force, which is to be estimated, is not simple. For example, one can expect that the strong force is easier to identify and with more confidence due to the larger cell displacement. On the other hand, the force field with a strong average value removes cells faster from the visual field and this ultimately results in fewer data points that are processed by the estimator. Fewer data points usually result in a higher uncertainty of the force estimation. Here, we propose a method for the uniform force field estimation based on an insight into the relation of the estimation uncertainty and trajectory lengths.

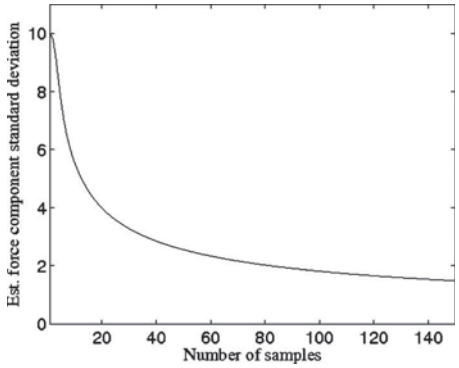


Fig. 7 The standard deviation of the constant force estimation component; the fixed-interval smoother parameter $\sigma_F=0$

Here, we deal with the uniform force field, i.e., $\mathbf{F}(\mathbf{x})=\mathbf{F}$ and $\mathbf{f}^i(t)=\mathbf{F}$ for any cell i, and the model (Eq. (5)) with parameters $c=1$, $\sigma=10$, $\mathbf{W}=\text{diag}\{1,1,1\}$, and $\boldsymbol{\Theta}=\text{diag}\{2.5^2,2.5^2,2.5^2\}$ that are identified from in vivo experimental data. The diagram in Fig. 7 shows the square root of the diagonal terms $[\mathbf{S}]_{11}=[\mathbf{S}]_{22}=[\mathbf{S}]_{33}$ of matrix $\mathbf{S}$, which is the submatrix of the Kalman filter covariance matrix $\mathbf{P}$ corresponding only to the force components, i.e.,

$$[\mathbf{S}]_{ij}=[\mathbf{P}]_{2D+i,2D+j}, \quad i,j=1,2,3, \quad D=3 \tag{14}$$

All these matrices can be computed based on the model parameters and $\sigma_F=0$. The diagonal terms $[\mathbf{S}]_{ii}$ and $i=1,2,3$ are equal due to the symmetry of our model regarding the change of coordinates.

Based on Fig. 7, we can see that the optimal smoother with parameter $\sigma_F=0$ (constant force assumption) cannot shrink the standard deviation of the force estimation below 10% of initial uncertainity for less than 100 samples of the trajectory. On the other hand, the cell trajectory length from experiments rarely goes over 100 samples. The typical number of samples is between 15 and 35. For this reason, and in order to increase the precision of the force estimation, it is beneficial to rely on the data across observed cell trajectories.

In the case of a constant force, estimations obtained for different trajectories can be integrated based on the maximum likelihood approach. Let us assume that we have applied the optimal force estimator with $\sigma_F=0$ to obtain the sequence of force estimations $\hat{\mathbf{F}}^i$, $i=1,2,\ldots$, and their covariance matrices $\mathbf{S}^i$. Under the assumption of the linear model (Eq. (5)) and the Kalman filter theory [15], the distribution of the estimated force is Gaussian so that the joint likelihood of the estimated values is (see Appendix D)

$$\mathcal{L}=\prod_{i=1,2,\ldots}\mathcal{N}(\hat{\mathbf{F}}^i-\bar{\mathbf{F}},\mathbf{S}^i) \tag{15}$$

where $\mathcal{N}(\cdot,\mathbf{S}^i)$ is a Gaussian distribution with a zero vector mean value and covariance matrix $\mathbf{S}^i$. The vector $\bar{\mathbf{F}}$ denotes the expected value of the constant force based on data from all cell trajectories. This expected value can be found by minimizing the log of likelihood $\mathcal{L}$, which results in the weighted average

$$\bar{\mathbf{F}}=\left[\sum_{i=1,2,\ldots}[\mathbf{S}^i]^{-1}\right]^{-1}\left[\sum_{i=1,2,\ldots}[\mathbf{S}^i]^{-1}\hat{\mathbf{F}}^i\right] \tag{16}$$

We can use the optimal force estimator with $\sigma_F=0$ and the above expression when we expect a uniform force field inside the visual field. However, we use the same expression for the estimation of a nonuniform force field. We will discuss this in the following section.

7 Nonuniform Field Estimation

For the estimation of the force field $\mathbf{F}(\mathbf{x})$ that influences cell motility, we propose a two-step strategy: first, to estimate forces $\mathbf{f}^i(k)$ that influence the cell motility at each observed cell position and second, to integrate all estimated values into a map that represents the force field, taking into account that the field $\mathbf{F}(\mathbf{x})$ is stationary in time.

From the results of the previous sections, we can see that we should assume the intensity of the variation of the forces $\mathbf{f}^i(k)$ and then properly tune the smoothing parameter σ_F. To improve the consistency of our estimation, instead of starting from $\mathbf{f}^i(k)$ as zero vectors, we estimate first the force $\bar{\mathbf{F}}$ in the same way that we estimate the constant force in the previous section. Then, we use this vector as an initial guess $\hat{\mathbf{f}}^i(0)$ for the estimation of the force along the ith cell trajectory at the kth time point $\hat{\mathbf{f}}^i(k)$. The motivation for this comes from the fact that the force $\bar{\mathbf{F}}$ is the best constant force that can describe all cell trajectories; thus, it is sensible to use it as an initial guess for $\hat{\mathbf{f}}^i(0)$ and all $i=1,2,\ldots$. With this guess and the smoothing algorithm, we can compute the estimates of the forces $\hat{\mathbf{f}}^i(k)$ at the cell positions $\hat{\mathbf{x}}^i(k)$ that we use in the second step.

Each estimate $\hat{\mathbf{f}}^i(k)$ corresponds to the cell position $\hat{\mathbf{x}}^i(k)$; in other words, we make identification that the force field estimation $\hat{\mathbf{F}}(\hat{\mathbf{x}}^i(k))=\hat{\mathbf{f}}^i(k)$. However, to build a consistent estimation of the force field, we also assume that each cell is influenced by the same force at the same position in the space. Taking this into account, we can say that for $\hat{\mathbf{x}}^j(k_m)\approx\hat{\mathbf{x}}^i(k_n)$ and obviously, $k_m\neq k_n$, we have $\hat{\mathbf{F}}(\hat{\mathbf{x}}^j(k_m))\approx\hat{\mathbf{F}}(\hat{\mathbf{x}}^i(k_n))$.

To estimate the field, we divide our visual field into R rectangular regions. We denote each of these regions as Ω_r, with $r=1,2,\ldots,R$ being the unique index of the region. Moreover, we define the set $\mathbf{In}(\Omega_r)$ as

$$\mathbf{In}(\Omega_r)=\{(i,k)|\hat{\mathbf{x}}^i(k)\in\Omega_r\} \tag{17}$$

Then, the force field $\mathbf{F}(\mathbf{x})$ for each region r, i.e., for $x\in\Omega_r$, can be estimated as the weighted average

$$\hat{\mathbf{F}}(\mathbf{x}\in\Omega_r)=\left[\sum_{(i,k)\in\mathbf{In}(\Omega_r)}[\mathbf{S}^i(k)]^{-1}\right]^{-1}\left[\sum_{(i,k)\in\mathbf{In}(\Omega_r)}[\mathbf{S}^i(k)]^{-1}\hat{\mathbf{f}}^i(k)\right] \tag{18}$$

This expression is identical to Eq. (16), except for the summation and product that go over all the couples (i,k) from the set $\mathbf{In}(\Omega_r)$ of the specific region Ω_r (see Appendix D).

We apply the described approach for the estimation of the force field to 40 cell trajectories generated from the model (Eq. (5)), $\sigma=10$, $c=1$, $\mathbf{W}=\text{diag}\{1,1,1\}$, and $\boldsymbol{\Theta}=\text{diag}\{2.5^2,2.5^2,2.5^2\}$, with the same geometrical constraints as in the previous section. In the example, we divide the visual field into $R=100$ rectangular regions of 20×15 μm^2 and generate the data assuming the force field

$$\mathbf{F}(\mathbf{x})=\begin{bmatrix}5\\5\sin(\pi x/100)\\0\end{bmatrix} \tag{19}$$

To include the fact that experimental data usually neglect short cell trajectories, we generated 200 cell trajectories and then randomly chose 40 cell trajectories, which had more than 16 points. Samples of the generated tracks are depicted in Fig. 2 in such a way that the first point of each trajectory is translated to the origin (0,0).

The result of our approach to force field estimation is presented in Fig. 8. The red arrows represent the force $\bar{\mathbf{F}}$ used as the initial condition for the estimation of $\mathbf{f}^i(k)$. To compute the field estimate

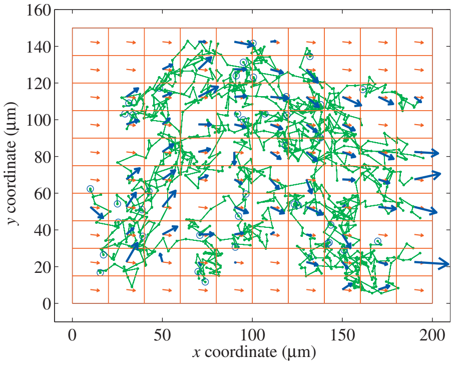


Fig. 8 Result of the force field estimation: the cell trajectories (green), the constant force $\bar{\mathbf{F}}$ (red arrows), the smoother estimation $\hat{\mathbf{F}}$ (blue thick arrows), and the interval smoother parameter σ_F=0.1. The true field is defined by expression (19).

$\hat{\mathbf{F}}$, we use a decreasing sequence of parameter σ_F, starting from considerable large $\sigma_F=10$, to obtain a series of estimations. When the region Ω_r contains less than three trajectory points, we skip averaging the forces estimated in the region, assuming that we have little data for averaging. For all regions Ω_r with more than three trajectory points, we compute the average $\hat{\mathbf{F}}$ and depict it with a blue thick arrow. The largest value of σ_F for which the estimations are consistent across the data generated from different simulation runs is $\sigma_F=0.1$. The blue arrows visualize the estimated force field from one of the simulation generated data sets.

The quality of the field estimation depends on cell trajectories; generally, the more recorded trajectories in the square region Ω_r, the more estimates per region available for averaging. In our case, the force component along the x axis is constant 5 (see Eq. (19)). The y component changes from 0 at points with $x=0$ to 5 at points $x=50$ and then back to 0 at points $x=100$. Then, this force component changes its sign and becomes negative, −5, at points with $x=150$ and again 0 at $x=200$. The force estimations that we acquired in Fig. 8 follow the above-described changes and give an insight into the field shape, although without great precision. The constant force $\bar{\mathbf{F}}$ (red arrows), which estimates the average component of forces in the visual field, is reliably estimated.

8 Conclusions

Following a technological breakthrough in imaging, we are now able to observe the evolution of in vivo immune system interactions. In this work, we have developed an approach for the estimation of the force field influencing cell motility. For this approach, we used the optimal smoother estimator of the forces and formed a map that visualizes the force field based on its estimations. We did not impose any analytical constraints on the force field. We introduced a spatial correlation between the forces by integrating estimations over a grid of rectangular regions. Consequently, the force field is estimated and visualized by the average forces assigned to each rectangular region.

Using the proposed method, we are confident to estimate the intensity and direction of the major constant component of the force field. We also find that the directions of the force field are satisfactorily identified. In order to apply the method for the analysis of imaging data routinely, the rapid adjustment of the estimator parameters is important. For this reason, we need further consideration of real data, including data collected from different immune system interactions and different organs.

Estimation of the forces that influence motility of immune system cells provides an additional powerful tool for the analysis of two-photon microscopic data. Our results show that real-time video images are sources of data, enabling visualization, which can be used for studying intriguing spatiotemporal phenomena inside immune system organs. We hope that our approach and similar approaches can fill the existing gap between contemporary technology and quantitative data analyses present in the field.

Acknowledgment

This work was supported by the Army Research Office under Short Term Innovative Research (STIR) Grant Nos. W911NF-07-R0003 and W911NF-08-1-0106.

Appendix A

For the illustration, we will derive the cell displacement for the case $D=2$. In the absence of the force $\mathbf{f}^i=0$, we can derive that the covariance of the x and y components of the position is

$$\Sigma_{xx}(t) = \Sigma_{yy}(t) = \Sigma(t) = 2\sigma^2\left(\frac{t}{2c^2} - \frac{3}{4c^3} + \frac{1}{c^3}e^{-ct} - \frac{1}{4c^3}e^{-2ct}\right) \tag{A1}$$

The probability density function of the position (x,y) is Gaussian and by introducing the radial displacement $\delta=\sqrt{x^2+y^2}$, we find that the probability density function of the displacement δ and the corresponding mean value are

$$p(\delta) = \frac{\delta}{\sigma^2}e^{-\delta^2/2\Sigma(t)}, \quad E\{\delta(t)\} = \sqrt{\Sigma(t)\frac{\pi}{2}} \tag{A2}$$

The mean value scales with $\sqrt{t}$ for $t\rightarrow\infty$.

Appendix B

For the motility model in two ($D=2$) or three ($D=3$) dimensions, the matrices are

$$\mathbf{A_c} = \begin{bmatrix} \mathbf{0}_{D\times D} & \mathbf{I}_{D\times D} \\ \mathbf{0}_{D\times D} & -c\mathbf{I}_{D\times D} \end{bmatrix}, \quad \mathbf{L_c} = \begin{bmatrix} \mathbf{0}_{D\times D} \\ \sigma\mathbf{I}_{D\times D} \end{bmatrix}$$

$$\mathbf{B_c} = \begin{bmatrix} \mathbf{0}_{D\times D} \\ \mathbf{I}_{D\times D} \end{bmatrix}, \quad \mathbf{C_c} = \begin{bmatrix} \mathbf{I}_{D\times D} & \mathbf{0}_{D\times D} \end{bmatrix} \tag{B1}$$

Let us introduce the transition matrix

$$\mathbf{\Phi}(t_k,t) = e^{\mathbf{A_c}(t_k-t)} \tag{B2}$$

Then, the state covariance of the continuous model

$$\dot{\mathbf{X}} = \mathbf{A_c}\mathbf{X} + \mathbf{L_c}\mathbf{w}(t) \tag{B3}$$

$$\mathbf{Y} = \mathbf{C_c}\mathbf{X} + \theta(t) \tag{B4}$$

sampled at regularly sampled time instants t_k, $t_{k+1}-t_k=\Delta$ is [15]

$$\mathbf{P_{k+1}} = \mathbf{\Phi}(t_{k+1},t_k)\mathbf{P}_k\mathbf{\Phi}(t_{k+1},t_k)^T + \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1},t)\mathbf{L_c}\mathbf{W}\mathbf{L_c}^T\mathbf{\Phi}(t_{k+1},t)^T dt \tag{B5}$$

Therfore, the regularly sampled continuous-time stochastic model (Eq. (B3)), with constant matrices $\mathbf{A_c}$, $\mathbf{L_c}$, and $\mathbf{W}=\mathbf{I}$ being the covariance of the vector $\mathbf{w}$, can be represented as the discrete-time stochastic process

$$\mathbf{X}(k+1) = \mathbf{A_d}\mathbf{X}(k) + \mathbf{L_d}\mathbf{w} \tag{B6}$$

$$\mathbf{Y}(k) = \mathbf{C_d}\mathbf{X}(k) + \theta(k) \tag{B7}$$

where

$$\mathbf{A_d} = e^{\mathbf{A_c}\Delta}, \quad \mathbf{L_d}\mathbf{L_d}^T = \int_0^\Delta e^{\mathbf{A_c}\Delta}\mathbf{L_c}(e^{\mathbf{A_c}\Delta}\mathbf{L_c})^T dt, \quad \mathbf{C_d} = \mathbf{C_c} \tag{B8}$$

To compute samples of the model (Eqs. (B6) and (B7)), the

Kalman filter, and the smoother, we only need the matrix $\mathbf{L_d}\mathbf{L_d}^T$. Therefore, we do not consider the problem of computing the matrix $\mathbf{L_d}$.

In the case of matrices (Eq. (B8)), we have

$$\mathbf{A_d} = \begin{bmatrix} \mathbf{I}_{D\times D} & \frac{1-e^{-c\Delta}}{c}\mathbf{I}_{D\times D} \\ 0 & e^{-c\Delta}\mathbf{I}_{D\times D} \end{bmatrix}, \quad \mathbf{L_d}\mathbf{L_d}^T = \sigma^2 \begin{bmatrix} \left(-\frac{e^{-2c\Delta}}{2c^3} + \frac{2e^{-c\Delta}}{c^3} + \frac{\Delta}{c^2} - \frac{3}{2c^3}\right)\mathbf{I}_{D\times D} & \left(\frac{e^{-2c\Delta}}{2c^2} - \frac{e^{-c\Delta}}{c^2} + \frac{1}{2c^2}\right)\mathbf{I}_{D\times D} \\ \left(\frac{e^{-2c\Delta}}{2c^2} - \frac{e^{-c\Delta}}{c^2} + \frac{1}{2c^2}\right)\mathbf{I}_{D\times D} & \left(-\frac{e^{-2c\Delta}}{2c} + \frac{1}{2c}\right)\mathbf{I}_{D\times D} \end{bmatrix} \tag{B9}$$

In the case of the augmented state discrete-time model (Eq. (7)), the matrices $\mathbf{A}$ and $\mathbf{L}$ are defined by expression (B8), where $\mathbf{A} = \mathbf{A_d}$, $\mathbf{L} = \mathbf{L_d}$, $\mathbf{C} = [\mathbf{C_c} \ \ \mathbf{0}_{D\times D}]$ and

$$\mathbf{A_c} = \begin{bmatrix} \mathbf{0}_{D\times D} & \mathbf{I}_{D\times D} & \mathbf{0}_{D\times D} \\ \mathbf{0}_{D\times D} & -c\mathbf{I}_{D\times D} & \mathbf{I}_{D\times D} \\ \mathbf{0}_{D\times D} & \mathbf{0}_{D\times D} & \mathbf{0}_{D\times D} \end{bmatrix}, \quad \mathbf{L_c} = \begin{bmatrix} \mathbf{0}_{D\times D} & \mathbf{0}_{D\times D} \\ \sigma\mathbf{I}_{D\times D} & \mathbf{0}_{D\times D} \\ \mathbf{0}_{D\times D} & \sigma_F\mathbf{I}_{D\times D} \end{bmatrix} \tag{B10}$$

This results in

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{D\times D} & \frac{1-e^{-c\Delta}}{c} & -\frac{1-e^{-c\Delta}}{c^2} + \frac{\Delta}{c} \\ 0 & e^{-c\Delta} & \frac{1-e^{-c\Delta}}{c} \\ 0 & 0 & 1 \end{bmatrix} \text{ and}$$

$$\mathbf{L}\mathbf{L}^T = \begin{bmatrix} \Lambda_{11}\mathbf{I}_{D\times D} & \Lambda_{12}\mathbf{I}_{D\times D} & \Lambda_{13}\mathbf{I}_{D\times D} \\ \Lambda_{12}\mathbf{I}_{D\times D} & \Lambda_{22}\mathbf{I}_{D\times D} & \Lambda_{23}\mathbf{I}_{D\times D} \\ \Lambda_{13}\mathbf{I}_{D\times D} & \Lambda_{23}\mathbf{I}_{D\times D} & \Lambda_{33}\mathbf{I}_{D\times D} \end{bmatrix} \tag{B11}$$

where

$$\Lambda_{11} = \sigma_F^2\left(\frac{e^{-2c\Delta}}{2c^4} - \frac{e^{-c\Delta}}{c^4} + \frac{\Delta e^{-c\Delta}}{c^3} + \frac{\Delta^2}{2c^2} - \frac{\Delta}{c^3}\right) + \sigma^2\left(-\frac{e^{-2c\Delta}}{2c^3} + \frac{2e^{-c\Delta}}{c^3} + \frac{\Delta}{c^2}\right) \tag{B12}$$

$$\Lambda_{12} = \sigma_F^2\left(\frac{2e^{-2c\Delta}}{c^4} - \frac{e^{-c\Delta}}{c^4} + \frac{\Delta e^{-c\Delta}}{c^3} + \frac{\Delta^2}{2c^2} - \frac{\Delta}{c^3} + \frac{3}{c^4}\right) + \sigma^2\left(\frac{1-e^{-c\Delta}}{c^2} - \frac{1-e^{-2c\Delta}}{2c^2}\right) \tag{B13}$$

$$\Lambda_{13} = \sigma_F^2\left(\frac{1-e^{-c\Delta}}{c^3} + \frac{\Delta^2}{2c} - \frac{\Delta}{c^2}\right), \quad \Lambda_{23} = \sigma_F^2\left(-\frac{1-e^{-c\Delta}}{c^2} + \frac{\Delta}{c}\right) \tag{B14}$$

$$\Lambda_{22} = \sigma_F^2\left(-\frac{e^{-2c\Delta}}{2c^3} + 2\frac{e^{-c\Delta}}{c^3} + \frac{\Delta}{c^2} - \frac{3}{2c^3}\right) + \sigma^2\frac{1-e^{-2c\Delta}}{2c}, \quad \Lambda_{33} = \sigma_F^2\Delta \tag{B15}$$

Appendix C

The Kalman filter iteration is described by the following equations [15]:

$$\hat{\mathbf{X}}_{\mathbf{F}}^{i-}(k+1) = \mathbf{A}\hat{\mathbf{X}}_{\mathbf{F}}^{i}(k) \tag{C1}$$

$$\mathbf{P}_{\mathbf{F}}^{i-}(k+1) = \mathbf{A}\mathbf{P}_{\mathbf{F}}^{i}(k)\mathbf{A}^T + \mathbf{L}^T\mathbf{L} \tag{C2}$$

$$\hat{\mathbf{X}}^{i}(k) = \hat{\mathbf{X}}_{\mathbf{F}}^{i-}(k) + \mathbf{K_{KF}}[\mathbf{y}^i(k) - \mathbf{C}\hat{\mathbf{X}}_{\mathbf{F}}^{i-}(k)] \tag{C3}$$

$$\mathbf{P}_{\mathbf{F}}^{i}(k) = [\mathbf{I} - \mathbf{K_{KF}}\mathbf{C}]\mathbf{P}_{\mathbf{F}}^{i-}(k) \tag{C4}$$

$$\mathbf{K_{KF}} = \mathbf{P}_{\mathbf{F}}^{i-}(k)\mathbf{C}^T[\mathbf{C}\mathbf{P}_{\mathbf{F}}^{i-}(k)\mathbf{C}^T + \mathbf{\Theta}]^{-1} \tag{C5}$$

The first equation (Eq. (C1)) is usually referred as the prediction step because it computes the model base prediction for the following time point. Similarly, Eq. (C2) is the model based prediction for the covariance matrices. Both of these values are computed starting from the initial values of $\hat{\mathbf{X}}_{\mathbf{F}}^{i}(0)$ and $\mathbf{P}_{\mathbf{F}}^{i}(0)$. Equation (C3) is the so-called update step, because it computes the optimal estimation $\hat{\mathbf{X}}^{i}(k)$ based on the prediction $\hat{\mathbf{X}}^{i-}(k)$ and observed data $\mathbf{y}^i(k)$. The Kalman filter gain $\mathbf{K_{KF}}$ depends on the prediction covariance matrix $\mathbf{P}_{\mathbf{F}}^{i-}(k)$, which is the covariance matrix of the estimated state vector $\hat{\mathbf{X}}_{\mathbf{F}}^{i-}(k)$.

Appendix D

In the case of the uniform field estimation, the estimations $\hat{\mathbf{F}}^i$ and $\hat{\mathbf{F}}^j$, $i \neq j$, resulting from trajectories of two different cells can be considered as two independent estimates of the force $\mathbf{F}$. Because of that, their joint probability density function is a product of their individual probability density functions, which are Gaussian according to the theory [15]. Thus, we can say that the likelihood of estimations from all available cell trajectories is

$$\mathcal{L} = \prod_{i=1,2,\ldots} \mathcal{N}(\hat{\mathbf{F}}^i - \bar{\mathbf{F}}, \mathbf{S}^i) \tag{D1}$$

where $\bar{\mathbf{F}}$ is the force $\mathbf{F}$ expected value. Maximization of this likelihood in order to find the estimation $\bar{\mathbf{F}}$ is equivalent to the minimization of its negative likelihood

$$-\log\mathcal{L} = \frac{1}{2}\sum_{i=1,2,\ldots}(\hat{\mathbf{F}}^i - \bar{\mathbf{F}})^T[\mathbf{S}^i]^{-1}(\hat{\mathbf{F}}^i - \bar{\mathbf{F}}) + C \tag{D2}$$

where C includes all terms independent on $\bar{\mathbf{F}}$. Derivatives of this expression regarding $\bar{\mathbf{F}}$ are equal to zero at the minimum, i.e., $\Sigma_{i=1,2,\ldots}[\mathbf{S}^i]^{-1}(\hat{\mathbf{F}}^i - \bar{\mathbf{F}}) = \mathbf{0}$, and from this we can conclude that

$$\bar{\mathbf{F}} = \left[\sum_{i=1,2,\ldots}[\mathbf{S}^i]^{-1}\right]^{-1}\left[\sum_{i=1,2,\ldots}[\mathbf{S}^i]^{-1}\hat{\mathbf{F}}^i\right] \tag{D3}$$

In the case of the time-stationary nonuniform field $\mathbf{F}(\mathbf{x})$, the force estimations along the two different trajectories $\hat{\mathbf{f}}^i(k_n)$ and $\hat{\mathbf{f}}^j(k_m)$, $i \neq j$ are independent random Gaussian vectors. Assuming that $\mathbf{F}(\mathbf{x})$ is constant over the square region Ω_r and neglecting the correlation among estimates from the same cell trajectory, i.e., between $\hat{\mathbf{f}}^i(k_m)$ and $\hat{\mathbf{f}}^i(k_n)$, $k_m \neq k_n$, we can write the likelihood of the force inside the region $\mathbf{F}(\mathbf{x} \in \Omega_r)$ as

$$\mathcal{L} = \prod_{(i,k) \in \mathbf{In}_{\Omega_r}} \mathcal{N}(\hat{\mathbf{f}}^i(k) - \hat{\mathbf{F}}(\mathbf{x} \in \Omega_r), \mathbf{S}^i(k)) \tag{D4}$$

where $\hat{\mathbf{F}}(\mathbf{x} \in \Omega)$ is the expected value of the force in square region Ω_r, and $\mathbf{In}_{\Omega_r}$ is defined by Eq. (17). Using this likelihood and applying steps (D1)–(D3), we can derive Eq. (18).

References

[1] Miller, M. J., Wei, S. H., Parker, I., and Cahalan, M. D., 2002, "Two-Photon Imaging of Lymphocyte Motility and Antigen Response in Intact Lymph Node," Science, **296**, pp. 1869–1873.

[2] Mempel, T. R., Henrickson, S. E., and von Andrian, U. H., 2004, "T-Cell Priming by Dendritic Cells in Lymph Nodes Occurs in Three Distinct Phases," Nature (London), **427**, pp. 154–159.

[3] Sumen, C., Mempel, T. R., Mazo, I. B., and von Andrian, U. H., 2004, "Intravital Microscopy: Visualizing Immunity in Context," Immunity, **21**, pp. 315–329.

[4] Milutinović, D., and Lima, P., 2007, *Cells and Robots: Modeling and Control of Large-Size Agent Populations*, Springer, New York.

[5] Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M., 2001, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," IEEE Trans. Rob. Autom., **17**(3), pp. 229–241.

[6] Thrun, S., Burgard, W., and Fox, D., 2005, *Probabilistic Robotics, Intelligent Robotics and Autonomous Agents*, MIT, Cambridge, MA.

[7] Bourne, H. R., and Weiner, O., 2002, "A Chemical Compass," Nature (London), **419**, p. 21.

[8] Comer, F. I., and Parent, C. A., 2002, "PI 3-Kinases and PTEN: How Opposites Chemoattract," Cell, **109**, pp. 541–544.

[9] Onsum, M., and Rao, C. V., 2007, "A Mathematical Model for Neutrophil Gradient Sensing and Polarization," PLOS Comput. Biol., **3**, pp. 436–459.

[10] Leonard, N., and Fiorelli, E., 2001, "Virtual Leaders, Artificial Potentials and Coordinated Control of Groups," Proceedings of the IEEE International Conference on Decision and Control, pp. 2968–2973.

[11] Tanner, H. G., Jadbabaie, A., and Pappas, G. J., 2007, "Flocking in Fixed and Switching Networks," IEEE Trans. Autom. Control, **52**(5), pp. 863–868.

[12] Olfati-Saber, R., 2006, "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," IEEE Trans. Autom. Control, **51**(3), pp. 401–420.

[13] Kumar, M., Milutinovíc, D., and Garg, D., 2008, "Role of Stochasticity in Self-Organization of Robotic Swarms," Proceedings of the American Control Conference, Seattle, WA.

[14] Astrom, K. J., and Wittenmark, B., 1990, *Computer-Controlled Systems: Theory and Design*, Prentice-Hall, Englewood Cliffs, NJ.

[15] Gelb, A., 1974, *Applied Optimal Estimation*, MIT, Cambridge, MA.

A.1.3 Spill Detection and Perimeter Surveillance via Distributed Swarming Agents

The following paper (on the next 9 pages) was co-authored by Guoxian Zhang, Gregory K. Fricke, and Devendra P. Garg, and was first published online September 2011 in IEEE/ASME Transactions on Mechatronics. The print version will appear February 2013 in Volume 18, Number 1, pages 121–129.

Spill Detection and Perimeter Surveillance via Distributed Swarming Agents

Guoxian Zhang, Gregory K. Fricke, and Devendra P. Garg

***Abstract*—The problem of perimeter detection and monitoring has a variety of applications. In this paper, a hybrid system of finite states is proposed for multiple autonomous robotic agents with the purpose of hazardous spill perimeter detection and tracking. In the system, each robotic agent is assumed to be in one of three states: searching, pursuing, and tracking. The agents are prioritized based on their states, and a potential field is constructed for agents in each state. For an agent in the tracking state, the agent's location and velocity as well as those of its closest leading and trailing agents are utilized to control its movement. The convergence of the tracking algorithm is analyzed for multiple spills under certain conditions. Simulation and experiment results show that with the proposed method, the agents can successfully detect and track the spills of various shapes, sizes, and movements.**

***Index Terms*—Autonomous agents, mobile agents, multirobot systems.**

I. Introduction

THE development of swarm robotics has emerged as a tool for mobile sensor networks in a variety of areas, such as environment monitoring [1], foraging [2], [3], target detection [4], and target tracking [5]. Two different kinds of control (centralized and decentralized) can be implemented for the robot group. In centralized control, all agents in the group are assumed to be able to share their information [6]. This kind of control suffers from computational burden if the number of agents is relatively large. More often, decentralized control is used for swarm robotics with information shared only among agents within a local network [1], [7]–[11]. In this paper, we propose a method for the problem of perimeter detection and tracking via swarm robots, which has a variety of applications such as forest fire surveillance [12], oil leakage tracking [13], and animal herd monitoring.

Some of the previous studies in this area have been reported in literature as follows. Casbeer *et al.* [12] proposed a decentralized multiple unmanned aerial vehicle (UAV) approach to monitor a forest fire. Each UAV flew along the perimeter of the fire in one direction, and then, reversed direction upon meeting another UAV. This movement continued and the movement of the UAVs formed a latency. This method could easily adapt to a change in the fire size and the number of UAVs needed during the process. Clark and Fierro [13] and Cruz *et al.* [14] utilized a hybrid control algorithm [15] to achieve the perimeter search and monitoring in an unknown environment; this method is susceptible to cases in which a single perimeter would split into multiple distinct perimeters. Bruemmer *et al.* [16] utilized the social potential field generated by different kinds of sensors on each agent to avoid collision and attract agents to the perimeter.

Related research in multiple robots is in formation control. Olfati-Saber [17], [18] proposed a method to control a flock while maintaining its geometric formation along the way. A potential function achieved local minima when the distances between each pair of agents in the flock reached a preset value. The stability of flocking of a group with fixed and dynamic topologies was studied by Tanner *et al.* in [19] and [20]. A social potential field was used to control the distances between agents, and each agent's velocity was controlled by the difference in the velocity from its neighbors. In these cases, the distance between each pair of agents converged to a preset value that is not suitable in cases where the desired distance between two neighbors changes with time.

In this paper, inspired by flocking agents, a control law is proposed for a group of agents whose purpose is to search for, detect, and track a hazardous spill in an unknown environment. Each agent is assumed to have limited and specific sensing and communication ranges. It is shown that the group can successfully detect the spill and track its boundary when the spill's location and size are changing. Simulation results showed excellent performance in tracking multiple spills that come together or split during the process. The control algorithm proposed in Section III is modified to fit the requirement of real experiments and the performance is verified with a group of robotic agents.

II. Problem Formulation

A group of autonomous mobile agents $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ is assumed to be initially deployed in the area $\mathcal{W}$. At time t, the configuration of agent i is represented as $\mathbf{q}_i(t) = [\mathbf{r}_i(t)^T, \theta_i(t)]^T$, where $\mathbf{r}_i(t) = [x_i(t), y_i(t)]^T$ denotes the position of $\mathcal{A}_i$ at time t and $\theta_i(t)$ denotes its orientation. Each agent is assumed to be equipped with a sensor whose field of view (FOV) is $\mathcal{S}_i$, and communication range is $\mathcal{M}_i$. For simplicity's sake, in our paper, we assume that both $\mathcal{S}_i$ and $\mathcal{M}_i$ are circles whose centers are located at $\mathbf{r}_i(t)$, and are of constant radii $r_{\mathcal{S}}$ and $r_{\mathcal{M}}$. Similar assumptions are commonly used by other researchers (for example, see [17] and [21]). A number of spills $\Omega_j(t)$ with the boundary $\partial\Omega_j(t)$ exist in $\mathcal{W}$. A spill $\Omega_j(t)$ is defined as a

Manuscript received September 30, 2010; revised March 9, 2011; accepted June 27, 2011. Date of publication September 12, 2011; date of current version September 12, 2012. Recommended by Technical Editor P. X. Liu. This work was supported in part by the Army Research Office under Grant W911NF-08-1-0106 and Grant W911NF-09-0307.

G. Zhang is with Microstrategy, Inc., Tysons Corner, VA 22182 USA (e-mail: zhangguoxian@gmail.com).

G. K. Fricke and D. P. Garg are with the Pratt School of Engineering, Duke University, Durham, NC 27708 USA (e-mail: gkf4@duke.edu; dpgarg@duke.edu).

Digital Object Identifier 10.1109/TMECH.2011.2164578

connected subspace in $\mathbb{R}^2$. Each agent is assumed to be able to measure the length of the spill boundary within its FOV. This length may be computed in a variety of ways. One way is to utilize pattern classification technology to detect the spill boundary from a visual sensor as implemented in [22]. Then, the detected spill boundary is represented by a series of points on it and a straight line segment is generated with each pair of adjacent points. The spill boundary length is approximated by summing the lengths of all the straight line segments. When $\partial\Omega_j(t)$ changes, it would be beneficial that the agent group can uniformly distribute along the spill boundary since the change of a part on $\partial\Omega_j(t)$ could be detected by an agent quickly.

A method is proposed to control the movement of each agent tracking a spill. Only the positions and linear speeds of an agent's closest leading and trailing neighbors along its way may be utilized to generate a potential field to control its movement. An influence distance along the spill boundary, denoted by L, controls how the closest neighbor affects an agent's potential field. This assumption guarantees the scalability of the proposed method and will be discussed in Section III.

The problem can be formulated as follows: given a group of agents $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ and their initial configurations $\mathbf{q}_1(0), \mathbf{q}_2(0), \ldots, \mathbf{q}_n(0)$, find the path for each agent i, where $i = 1, 2, \ldots, n$, such that $\forall\epsilon > 0$, there exists a time $T > 0$ and a spill index j, for $t > T$ agent i tracks $\Omega_j(t)$, and $\|\mathbf{r}_i(t), \mathbf{r}_{N(\mathbf{r}_i(t))}(t)\|_{\partial\Omega_j} = L_j(t)$ satisfies

$$\begin{cases} |L_j(t) - \dfrac{\|\partial\Omega_j(t)\|}{n_j(t)}| < \epsilon, & \text{if } n_j(t) > \dfrac{\|\partial\Omega_j(t)\|}{L} \\ L_j(t) >= L - \epsilon, & \text{if } n_j(t) \le \dfrac{\|\partial\Omega_j(t)\|}{L}. \end{cases} \tag{1}$$

Here, $\|a, b\|_{\partial c}$ is the distance from a to b along the curve ∂c in the forward direction; $N(\mathbf{r}_i(t))$ is the index of the closest neighbor of $\mathcal{A}_i$ in the forward direction; $n_j(t)$ is the number of agents that are tracking Ω_j at time t; and $\|\partial\Omega_j(t)\|$ is the length of $\partial\Omega_j(t)$. Furthermore, the velocities of agents tracking the same spill converge to the same value. Mathematically speaking, assume that agents i and j are tracking the same spill, $\forall\epsilon > 0, \exists T > 0$, such that $\forall t > T$, we have $|v_i(t) - v_j(t)| < \epsilon$. When there is no ambiguity, we simplify each variable without writing time explicitly; for example, $\partial\Omega_j$ represents $\partial\Omega_j(t)$.

The communication among the mobile agents can be represented as a set of connected undirected graphs, each of which can be defined as $\mathcal{G}_k(\mathcal{V}_k, \mathcal{E}_k)$, where $\mathcal{V}_k = \{a_{l_1}, a_{l_2}, \ldots, a_{l_{\ell k}}\}$ are the agents in $\mathcal{G}_k$, and $\mathcal{E}_k = \{(a_{l_i}, a_{l_j}) \in \mathcal{V}_k \times \mathcal{V}_k | a_{l_i}, a_{l_j} \in \mathcal{V}_k, a_{l_j} \in \mathcal{M}_{l_i}\}$. The information, such as a detected point of the spill boundary by an agent, is assumed to be able to be shared within the graph with no time delay. A typical example of the agent system with ten agents and one spill is shown in Fig. 1.

III. Solution

A hybrid hierarchical control technique is used to control the agents. Three behaviors are used for perimeter detection and tracking: searching, pursuing, and tracking. An agent switches among these states based on the agent and its local group members' situations. The algorithmic state transitions for an individual robot are shown in Fig. 2. This hybrid hierarchy is consistent with the embedded software principles of Alur *et al.* [15]. The system described herein exhibits architectural hierarchy in its basic construction; the controller employs behavioral hierarchy in its state transitions. In contrast to the hybrid controller of Clark and Fierro [13] and Cruz *et al.* [14], the controller presented here requires only a single tier of mode switching with no submodes or substates for obstacle or vehicle avoidance. The potential functions, developed next, allow a continuous calculation and application of control authority that is consistent with the goals of each state and yet robust to collision avoidance.

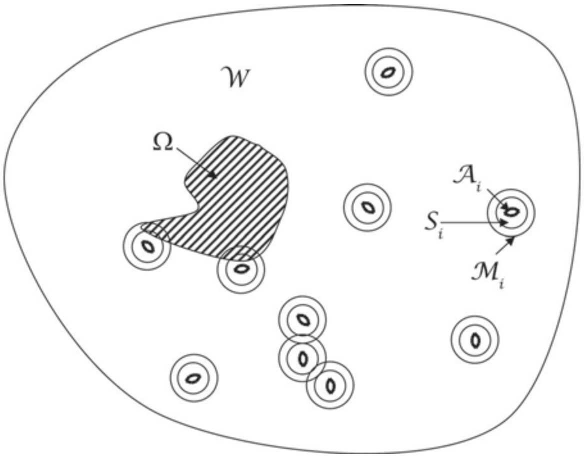
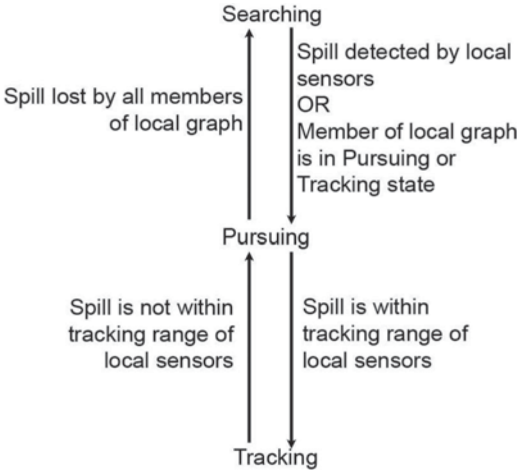


Fig. 1. Sample layout of the multiple agent system.



Searching
Spill lost by all members of local graph
Spill detected by local sensors OR Member of local graph is in Pursuing or Tracking state
Pursuing
Spill is not within tracking range of local sensors
Spill is within tracking range of local sensors
Tracking

Fig. 2. State transition diagram.

The agents are assumed to have a unicycle model in which

$$\begin{cases} \dot{x}_i = v_i \cos\theta_i \\ \dot{y}_i = v_i \sin\theta_i \\ \dot{v}_i = a_i \\ \dot{\theta}_i = \omega_i \end{cases} \tag{2}$$

where v_i is the linear velocity of the ith agent, a_i is its linear acceleration, and ω_i is its angular velocity. The control inputs of each agent are set up as $[u_i \;\; w_i]$, where $u_i = a_i$ and $w_i = \omega_i$.

A. Perimeter Search

In the proposed method, the robotic agents are initially randomly deployed in an unknown environment. The geometry of each agent is ignored in the system, i.e., two agents collide only when their positions are the same. To avoid collision, we assume that there is a repulsive potential field generated by each agent, denoted as U_i [23]

$$U_i(\mathbf{r}) = \begin{cases} \frac{1}{2}\eta_1\left(\frac{1}{\rho(\mathbf{r},\mathbf{r}_i)} - \frac{1}{\rho_0}\right)^2, & \text{if } \rho(\mathbf{r},\mathbf{r}_i) \le \rho_0 \\ 0, & \text{if } \rho(\mathbf{r},\mathbf{r}_i) > \rho_0 \end{cases} \tag{3}$$

where η_1 is a scaling parameter, ρ_0 is the influence distance of each agent, $\mathbf{r}_i$ is the ith agent position, and $\rho(\mathbf{r},\mathbf{r}_i)$ is the distance between $\mathbf{r}$ and $\mathbf{r}_i$.

The agents in different states are assumed to have different priority levels for avoidance. The agents in the searching state are assumed to be at the lowest level, and the agents in the tracking state are at the highest level. Let the index sets I_s, I_p, and I_t represent the indices of agents in searching, pursuing, and tracking states, respectively. Assume that an agent in the searching state is at $\mathbf{r}$, and let $N_s(\mathbf{r}) = \{i|\forall i,\ \mathbf{r} \in \mathcal{M}_i\}$ denote the indices of agents in which communication range $\mathbf{r}$ lies. When $N_s(\mathbf{r}) \neq \emptyset$, the following potential field is constructed for the agent:

$$U_s(\mathbf{r}) = \sum_{i \in N_s(\mathbf{r})} U_i(\mathbf{r}) \tag{4}$$

and the force implemented on the agent is computed as

$$F_s(\mathbf{r}) = -\nabla U_s(\mathbf{r}). \tag{5}$$

Here, $\|F_s(\mathbf{r})\|$ is utilized as control u_i for the agent, and ω_i is set at the value with which the orientation of the agent in the next time step is consistent with the direction of $F_s(\mathbf{r})$. Since the travel distance and search time are not considered in the objective of our problem, when $N_s(\mathbf{r}) = \emptyset$, a random search technique in discrete time with the input $[u_i^r(k), w_i^r(k)]$ is utilized to navigate the agent at $\mathbf{q}$ with searching state

$$\begin{aligned} u_i^r(k+1) &= u_i^r(k) + b(k)\Delta t \\ w_i^r(k+1) &= w_i^r(k) + c(k)\Delta t \end{aligned} \tag{6}$$

where $b(k)$ is a random variable having uniform distribution with the support assigned by the user, $c(k)$ is a random variable having normal distribution with a zero mean, and variance specified by the user, and Δt is the discrete time period. When another objective such as travel distance is included, collaboration between agents may be used to improve the search efficiency (e.g., see Franchi *et al.* [8]). Alternatively, task allocation such as that developed by Viguria and Howard [24] may be implemented.

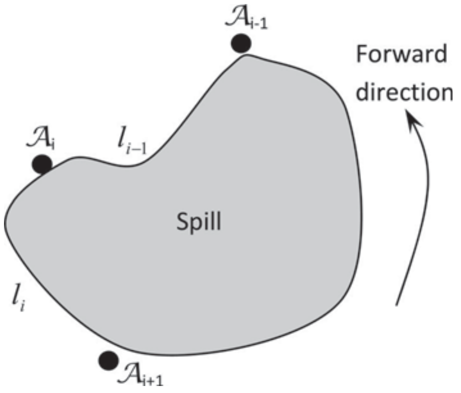


Fig. 3. Location, forward movement direction, and distance between adjacent mobile agents along the spill perimeter.

B. Pursuit

When a point lying on the boundary of the spill is within the ith agent's sensor FOV, a detection occurs. The ith agent changes to a pursuing state if a detection occurs either in its FOV or by a member of its local communication graph. The location of the detected point is denoted by $\mathbf{r}_d$. A potential field $U_d(\mathbf{r})$ is constructed for the agent moving toward $\mathbf{r}_d$. It is given by

$$U_d(\mathbf{r}) = \frac{1}{2}\eta_2\rho(\mathbf{r},\mathbf{r}_d)^2 \tag{7}$$

where η_2 is a scaling parameter and $\rho(\mathbf{r},\mathbf{r}_d)$ is the distance between $\mathbf{r}$ and $\mathbf{r}_d$. Let $N_p(\mathbf{r}) = \{i|\forall i \in I_p \cup I_t,\ \mathbf{r} \in \mathcal{M}_i\}$ denote the set of indices of agents in pursuing or tracking state, in which communication range $\mathbf{r}$ lies. The following potential field is constructed for the agent:

$$U_p(\mathbf{r}) = U_d(\mathbf{r}) + \sum_{i \in N_p(\mathbf{r})} U_i(\mathbf{r}). \tag{8}$$

The negative gradient $F_p(\mathbf{r})$ of the potential $U_p(\mathbf{r})$ is utilized to navigate the agent in pursuing state. In practice, we assume that there is a maximum velocity $v_{\max}$ to bound the movement of each agent.

C. Tracking the Spill Boundary

Assume that the agent i is in the pursuing state. When $\|\mathbf{r}_i - \mathbf{r}_d\| < d$, where d is a capture distance defined by the user, the agent changes into the tracking state. It will then move along the spill boundary with the interior of the spill on its left. The forward direction of the agents' movement is as shown in Fig. 3.

When an agent moves away from the forward direction, its velocity is negative. In this case, the unicycle model in (2) can still be used by simply adding π to the orientation. For agent i, l_i is defined as follows. Assume that at time t, the ith agent is tracking Ω_j. It will measure the length of the perimeter and positions of other agents in the tracking state within its FOV. Practically, this measurement may be performed by calibrated webcams (low cost, but low accuracy), laser rangefinders (higher cost and high accuracy), or a system such as that developed by Pugh *et al.* [25]. Then, if the closest neighbor in the forward direction, whose index is represented by $N(\mathbf{r}_i)$, is inside its FOV, set $l_i = \|\mathbf{r}_i, \mathbf{r}_{N(\mathbf{r}_i)}\|_{\partial\Omega_j}$; otherwise, l_i is unknown to the agent. Assume that $\partial\Omega_j$ can be represented by a function $s = f(\mathbf{r})$, where $s \in [0, \|\partial\Omega_j\|]$ indicates the length of the perimeter from a reference point to $\mathbf{r}$ along the forward direction. Write $f(\mathbf{r}_i)$

as s_i and $s(\mathbf{r}_{N(\mathbf{r}_i)})$ as s_{i+1}, then when l_i is known to the agent i, it can be computed as follows:

$$l_i = \begin{cases} s_{i+1} - s_i, & \text{if } s_{i+1} \geq s_i \\ s_{i+1} - s_i + \|\partial\Omega_j\|, & \text{if } s_{i+1} < s_i. \end{cases} \tag{9}$$

Define a virtual distance L_i between agent i and $N(\mathbf{r}_i)$ as

$$L_i = \begin{cases} l_i, & \text{if } l_i < L \\ L, & \text{if } l_i \geq L \text{ or } l_i \text{ is unknown} \end{cases} \tag{10}$$

where L is the influence distance between agent i and agent $N(\mathbf{r}_i)$, and is set as $L = r_{\mathcal{S}}$ in our paper.

Assuming that $L_0 = L_{n_j}$, i.e., the last agent wraps to the first agent, a potential function [26], [27] is proposed for agents i, which is given by

$$V_i(L_i, L_{i-1}) = k_1\Big(2ln(L_i) + \frac{L_{i-1}}{L_i}\Big) \tag{11}$$

where k_1 is a positive constant. When $L_i < L$ and $L_{i-1} < L$, using (9)–(11), the derivative of V_i with respect to s_i is

$$\frac{dV_i}{ds_i} = -k_1\left(\frac{1}{L_i} - \frac{L_{i-1}}{L_i^2}\right) \tag{12}$$

where V_i has a minimum at $L_i = L_{i-1}$. Including the control of velocity for the agents, the following controller is proposed to be used as the linear acceleration input of the ith agent:

$$u_i = \begin{cases} k_1\left(\frac{1}{L_i} - \frac{L_{i-1}}{L_i^2}\right) & \text{if } L_i < L,\, L_{i-1} < L \\ \quad +(v_{i+1} - 2v_i + v_{i-1}), & \\ k_1\left(\frac{1}{L} - \frac{L_{i-1}}{L^2}\right) + (v_{i-1} - v_i), & \text{if } L_i = L,\, L_{i-1} < L \\ k_1\left(\frac{1}{L_i} - \frac{L}{L_i^2}\right) + (v_{i+1} - v_i), & \text{if } L_i < L,\, L_{i-1} = L \\ 0, & \text{if } L_i = L,\, L_{i-1} = L. \end{cases} \tag{13}$$

From (12) and (13), it is evident that when L_i goes to zero, u_i approaches minus infinity. This guarantees that the ith agent does not surpass the $(i+1)$th agent, which implies that the order of the agents is maintained when they are moving.

The angular velocity of the ith agent is also controlled discretely. Assume that the change in distance traveled by the ith agent during Δt is Δs_i. The change in θ_i, denoted as $\Delta\theta_i$, is computed in different situations, as shown in Fig. 4.

If another kind of agent, denoted as $\mathcal{B}$, which has a much larger measurement range, (e.g., an airborne agent) is available to monitor the spill, while the multiple robotic agents are tracking the spill, its measurement can be used as a coarse measure of l_i. This measurement of l_i is likely to have a lot of associated noise due to the low resolution of the global measurement. In that case, (10) may be changed to

$$L_i = \begin{cases} l_i, & \text{if } l_i < L \\ \hat{l}_i, & \text{if } l_i \geq L \end{cases} \tag{14}$$

where $\hat{l}_i$ is the noisy measurement by agent $\mathcal{B}$. This case will be studied in our future study.

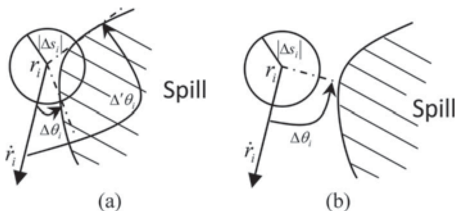


Fig. 4. Change in θ_i at each time step. (a) Movements of agent and spill have intersection; if $\Delta s_i > 0$, angular change is $\Delta\theta_i$; if $\Delta s_i < 0$, angular change is $\Delta'\theta_i$. (b) Movements of agent and spill have no intersection; agent moves toward the closest boundary point of the spill.

D. Stability Analysis

To investigate the system's stability, we consider a time $t > T$ in which we assume that all agents are in the tracking state, and the spills stop changing in size and location. Another assumption is that the number of agents is large enough so that each agent is connected with its leading and trailing neighbors during tracking. Without loss of generality, we relabel the agents so that the first n_1 agents track Ω_1, the (n_1+1)th–(n_1+n_2)th agents track Ω_2, and so on. Let $\mathbf{l} = [L_1, L_2, \ldots, L_n]^T$. The collective potential of the system can be given by

$$V(\mathbf{s}) = V(\mathbf{l}) = \sum_{j=1}^{p}\left(\sum_{i=n_0+\cdots+n_{j-1}+1}^{n_0+\cdots+n_j} V_i\right) \tag{15}$$

where $\mathbf{s} = [s_1, \ldots, s_n]^T$; p is the number of spills when the spills stop changing; n_0 equals zero; and $M_1, M_2, \ldots, M_p$ are the lengths of the p spill boundaries; $L_1, L_2, \ldots, L_n$ satisfies

$$\begin{cases} L_1 + L_2 + \cdots + L_{n_1} = M_1 \\ L_{n_1+1} + L_{n_1+2} + \cdots + L_{n_1+n_2} = M_2 \\ \cdots \\ L_{n_1+\cdots+n_{p-1}+1} + L_{n_1+\cdots+n_{p-1}+2} + \cdots + L_n = M_p. \end{cases} \tag{16}$$

The collective dynamics of the system is given by

$$\dot{\mathbf{s}} = \mathbf{v} \tag{17}$$

$$\dot{\mathbf{v}} = -\left[\frac{\partial V_1}{\partial s_1}, \frac{\partial V_2}{\partial s_2}, \ldots, \frac{\partial V_n}{\partial s_n}\right]^T - \mathcal{L}(\mathbf{s})\mathbf{v} \tag{18}$$

where $\mathbf{v} = [v_1, \ldots, v_n]^T$ and $\mathcal{L}$ is the graph Laplacian as described in [17] and [28].

By using Lagrange Multipliers, (15) has an extremum point at

$$\begin{cases} L_1 = L_2 = \cdots = L_{n_1} = \dfrac{M_1}{n_1} \\ L_{n_1+1} = L_{n_1+2} = \cdots = L_{n_1+n_2} = \dfrac{M_2}{n_2} \\ \cdots \\ L_{n_1+\cdots+n_{p-1}+1} = L_{n_1+\cdots+n_{p-1}+2} = \cdots = L_n = \dfrac{M_p}{n_p}. \end{cases} \tag{19}$$

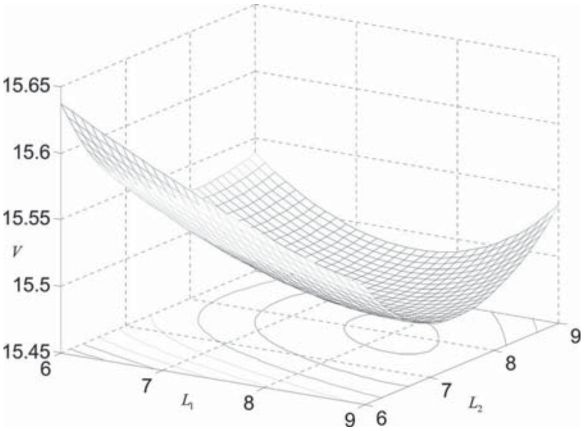


Fig. 5. Example of the collective potential value changing with L_1 and L_2.

Although it is analytically difficult to show that this point is the only minimum for (15), numerical simulations with MATLAB support this assumption. For example, when $n = 3$, $M = 24$, $p = 1$, and $k = 1$, the value of V is shown in Fig. 5, where the minimum of V appears at $L_1 = L_2 = L_3 = 8$ and satisfies (19). Other values of n, M, and p also satisfy (19) as verified by simulations.

Let $H(\mathbf{s}, \mathbf{v})$ represent the energy of the system, i.e.

$$H(\mathbf{s}, \mathbf{v}) = V(\mathbf{s}) + \frac{1}{2}\sum_{i=1}^{n} v_i^2. \tag{20}$$

Then,

$$\begin{aligned}\dot{H}(\mathbf{s}, \mathbf{v}) &= \nabla V(\mathbf{s})^T \mathbf{v} - \left[\frac{\partial V_1}{\partial s_1}, \dots, \frac{\partial V_n}{\partial s_n}\right]^T \mathbf{v} - \mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v} \\ &= \left[\frac{\partial V_{n_1}}{\partial s_1} + \frac{\partial V_2}{\partial s_1}, \dots, \frac{\partial V_{n-1}}{\partial s_n} + \frac{\partial V_{n-n_p+1}}{\partial s_n}\right] \mathbf{v} - \mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v} \\ &= \left\{\left[\frac{2}{L_{n_1}} - \frac{L_{n_1-1}}{L_{n_1}^2} - \frac{1}{L_2}, \dots, \frac{2}{L_{n-1}} - \frac{L_{n-2}}{L_{n-1}^2} - \frac{1}{L_{n-n_p+1}}\right]\mathbf{v}\right\} \\ &\quad - \mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v}\end{aligned} \tag{21}$$

where $\mathcal{L}$ has the form

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_1 & & & \\ & \mathcal{L}_2 & & \\ & & \ddots & \\ & & & \mathcal{L}_p \end{pmatrix} \tag{22}$$

which is a positive semidefinite block diagonal matrix with rank $n - p$, and $\mathcal{L}_j$ has dimension n_j. In general, $\dot{H}$ may be positive, because the potential function for agent i is based on only the closest leading and trailing neighbors.

Therefore, for agent i, no information on V_{i-1} and V_{i+1} is available. However, with bounded $\mathbf{v}$ and the control in (13), each L_i, where $i = 1, 2, \dots, n$, tends to be close to each other, and then, the first set of terms in curly brackets on the right side of (21) gets close to zero. Thus, the sign of $\dot{H}$ is dominated by $-\mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v}$ and is negative.

In this case, $\dot{H}$ equals zero when $\mathbf{v}$ is an eigenvector of $\mathcal{L}$ corresponding to eigenvalue zero. Then, $\mathbf{v}$ will be in the space with basis containing p vectors, denoted by $\mathbf{b}_i = [c_1, c_2, \dots, c_n]$, where

$$c_j = \begin{cases} 1, & n_0 + \dots + n_{i-1} + 1 < j \le n_0 + \dots + n_i \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

for $i = 1, 2, \dots, p$, and

$$\mathbf{v} = \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 + \dots + \lambda_p \mathbf{b}_p \tag{24}$$

where λ_i, $i = 1, 2, \dots, p$, is an arbitrary parameter that in this case represents the steady linear velocity of all agents on $\partial\Omega_i$. This shows that the velocities of all agents tracking the same spill converge to the same value. Since $\mathbf{v}$ is bounded, the convergent velocity is also bounded. When H achieves a minimum, V is also at its local minimum. From previous analysis

$$L_{n_1+\dots+n_{j-1}+1} = \dots = L_{n_1+\dots+n_j}, \qquad \text{for } j = 1, 2, \dots, p \tag{25}$$

is a local minimum for V. Hence, the system achieves the stable state with equal distance between adjacent agents and equal velocity for all agents tracking the same spill.

Consider a case that after all agents change to tracking state, spill Ω_1 does not have enough agents tracking along its boundary, i.e., $n_1 < \|\partial\Omega_1\|/L$. We claim that all agents tracking Ω_1 converge to the same velocity and the distance between each pair of adjacent agents converges to a value no less than L. The energy of the agents tracking Ω_1 is presented as

$$H_1(\mathbf{s}_1, \mathbf{v}_1) = V(\mathbf{s}_1) + \frac{1}{2}\sum_{i=1}^{n_1} v_i^2. \tag{26}$$

Similarly,

$$\begin{aligned}\dot{H}(\mathbf{s}_1, \mathbf{v}_1) = &\left[\frac{2}{L_{n_1}} - \frac{L_{n_1-1}}{L_{n_1}^2} - \frac{1}{L_2}, \dots,\right. \\ &\left.\times \frac{2}{L_{n_1-1}} - \frac{L_{n_1-2}}{L_{n_1-1}^2} - \frac{1}{L_1}\right]\mathbf{v}_1 - \mathbf{v}_1^T \mathcal{L}(\mathbf{s}_1)\mathbf{v}_1\end{aligned} \tag{27}$$

where $\mathbf{s}_1 = [s_1, \dots, s_{n_1}]$ and $\mathbf{v}_1 = [v_1, \dots, v_{n_1}]$. Without loss of generality, we assume that at time t, $\mathcal{L}_1$ is a block diagonal matrix containing two blocks $\mathcal{L}_{11}$ and $\mathcal{L}_{12}$. In view of the discussion before, if no connection between $\mathcal{L}_{11}$ and $\mathcal{L}_{12}$ holds, the velocity of agents in $\mathcal{L}_{11}$ converges to v^1 and the velocity of agents in $\mathcal{L}_{12}$ converges to v^2. If $v^1 \neq v^2$, the two connected graphs will connect with each other some time since they are tracking a perimeter. Then, (27) is likely to be negative and cause the energy of the system to decrease again until either the graph splits into several connected graphs or the velocity of all agents converges to the same value. Since different velocities of agents in various connected graphs cause a decrease of the system energy, the system energy achieves a local minimum when the agents within all connected graphs tracking Ω_1 have the same velocity. In this case, the first agent at the forward direction of each connected graph will have a leading virtual distance equal to L (i.e., the distance between this agent and its leading adjacent agent is larger than L), which causes the distance between adjacent agents within its graph to be L to have a minimal energy as well. Therefore, the distance between each pair of adjacent agents tracking Ω_1 is not less than L.

IV. Simulation and Experiment Results

In this section, simulations and experiments are described to test the performance of the proposed method based on different conditions. The influence of spill locations, size, and numbers as well as the agent group size is studied.

A. Simulation Results

In this section, we build a simulation to see how the agent group adapts to the change of the spill number as well as its location and size. In this environment, we assume that there are two spills at the beginning. As time goes on, the two spills merge into one spill, and then, again split into two spills. The number of agents is varied from 6 to 46, $\mathcal{W}$ is a square arena with sides of length 40 m, $r_{\mathcal{S}} = 5$ m, $r_{\mathcal{M}} = 8$ m, $v_{\max} = 8$ m/s, and $\Delta t = 0.01$ s. All simulation results are averaged over 30 runs. The result of one run with 26 agents is shown in Fig. 6.

From the time snapshots of the simulation process, we can see that the agents successfully search, pursue, and track the spills no matter if they are merged or split into pieces. After the spills stop changing, the velocities of agents tracking the same spill converge to the same value as well as maintaining constant distance between adjacent agents. Readers are referred to [29] for the results of examples with other spills.

Fig. 7 shows the average time required for all agents to reach tracking state as a function of the number of agents. Five separate simulations, using, respectively, 6, 16, 26, 36, and 46 agents, were run with all other parameters set to the same values used in the simulation, as shown in Fig. 6. The results show that as the number of agents increases, the group requires less time for all agents to change the tracking state. This is because with more agents, it is more likely that several agents will be connected in a local graph and that the possibility for an agent in the group to detect a spill is increased. This decreases the time that the agent group needs to explore the environment to detect spills. For more than 26 agents, the improvement in time decreases as the number of agents increases. The reason is that when the number of agents is large, most agents are connected in the local graph and immediately achieve the pursuing state as soon as one agent in the local graph detects a spill; adding more agents to the group does not decrease the time to detection as much as it does when the number of agents is small.

To demonstrate how the size of the sensor FOV can affect the process time for all agents to change the tracking state, another simulation is conducted utilizing 16 agents, with different $r_{\mathcal{S}}$ values of 1, 3, 5, 10, 15, 20, and 30. The results are shown in Fig. 8. The results show that as the sensor FOV increases, the needed time decreases. The rate of decrease tends to be small when $r_{\mathcal{S}}$ is larger than 15. The reason is that as $r_{\mathcal{S}}$ increases, the possibility for the robotic sensor to detect a spill increases; however, when $r_{\mathcal{S}}$ is large enough compared to the size of the workspace,

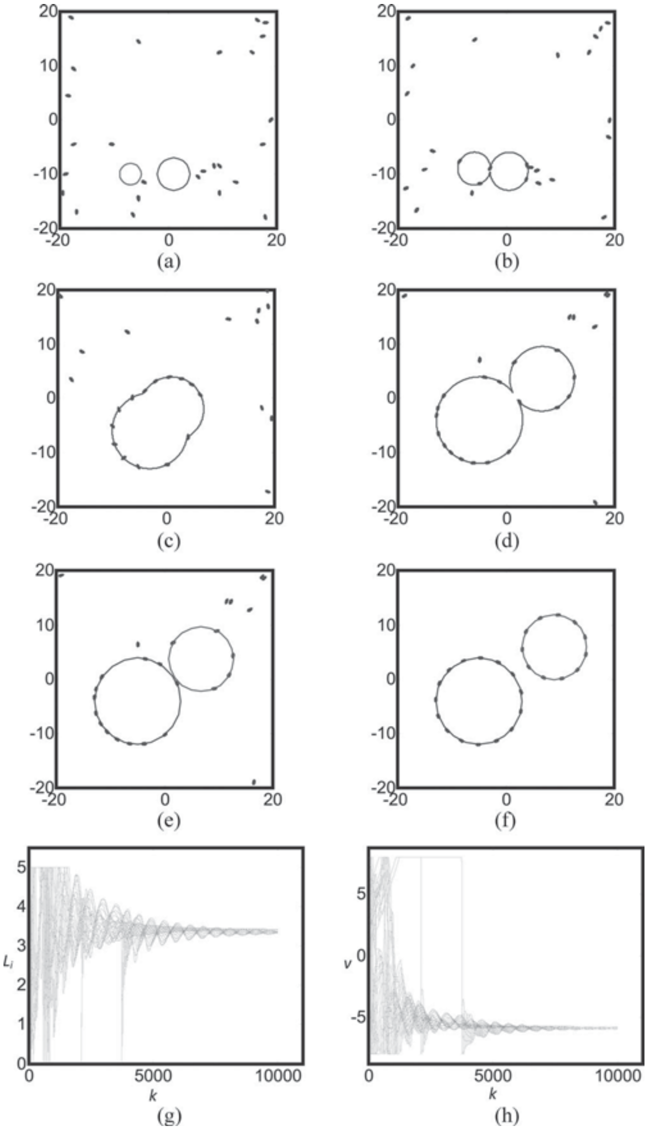


Fig. 6. Simulation result with changing number of spills. (a) Snapshot at time step one. (b) Snapshot at time step 52. (c) Snapshot at time step 400. (d) Snapshot at time step 682. (e) Snapshot at time step 690. (f) Snapshot at time step 5000. (g) Distance between each agent and the closest agent ahead of it. (h) Velocity of each agent.

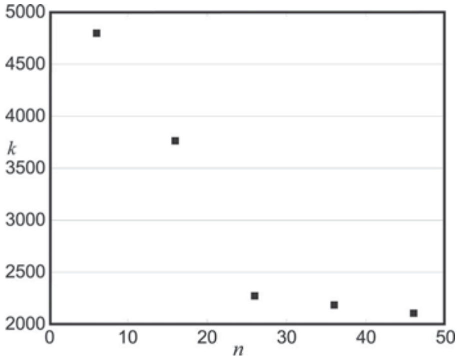


Fig. 7. Average time needed for all the agents to change the tracking state for different numbers of agents. The results are averaged over 30 runs.

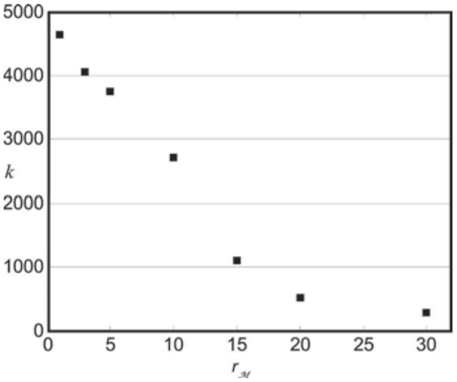


Fig. 8. Average time needed for all of the agents to change the tracking state under different sizes of sensor FOV. The results are averaged over 30 runs.

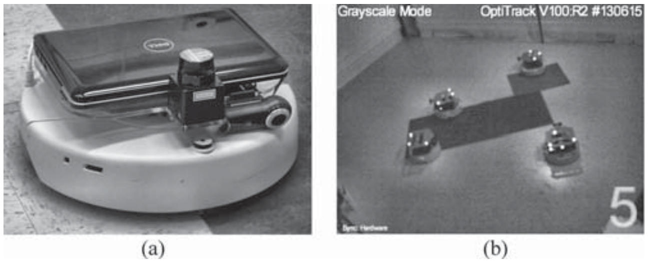


Fig. 9. (a) Experimental configuration of Create robot with Dell Mini10 netbook, Logitech Quickcam Pro 9000, and Hokuyo URG-04LX-UG01 scanning laser rangefinder. (b) Perspective from camera no. 5 within the system.

the agent could easily find the spill in a short time, and further increase of $r_{\mathcal{S}}$ will not shorten the search time significantly.

B. Experiment Results

Several experiments using real robots in a mixed simulation setting have been conducted. The agents in these experiments are iRobot Create robots. Dell Mini10 netbook computers are mounted to each robot, sending drive commands to and receiving sensor data from the Create base.

Each robot is equipped with reflective markers in a unique pattern for identification to allow tracking via NaturalPoint OptiTrack V100R2 cameras and customized Tracking Tools software on a centralized computer (see [30] and [31] for alternatives) . The tracking data, akin to "indoor GPS," are available to each individual robot via wireless ethernet (WiFi) from this central switchboard. The laboratory space is approximately 3 m $\times$ 6 m, which is well within WiFi range; use of the switchboard computer enables the simulation of limited communication range by allowing (respectively, blocking) the communication between robots that are inside (respectively, outside) a specified range.

The spills are simulated by covering an area of the floor with a material of contrasting color. The spill sensor on each robot is a set of four downward looking IR sensors mounted on the front bumper. The robot cannot detect the spill until it is already sufficiently close to enter tracking state. If no other spill sensor is equipped, the time spent in pursuit state in these experiments will be only one time step if the robot has directly detected the spill. The only case in which a robot remains in pursuit state for extended time is when spill detection occurs by another member of the local communication graph.

The downlooking IR sensors of the Create robot are thus used as the primary sensors for actually tracking the spill; when the spill is detected in these sensors, the robot switches into track state.[1]

A sample active-IR image of multiple robots captured by camera no. 5 is shown in Fig. 9(b). The markers are clearly visible in this full-frame image, as are the spill and robot shapes. The robots in this experiment are constrained to a plane; thus, the out-of-plane distance measurement and the roll and pitch orientation measurements are unnecessary (and should be unchanging). For more details, the reader is referred to [22].

For each experiment, the robots were distributed about the laboratory space away from the spill, with no particular arrangement or pattern. Results for two experiments are presented here although several different configurations of spills have been considered in this paper, including convex and nonconvex shapes, shapes with different areas, and contiguous or noncontiguous areas. Both experiments presented here utilized five robots in the configuration discussed; several other experiments were conducted with three or four robots, as well.

For the experiments reported here, the communication range is $r_{\mathcal{M}} = 2$ m, the avoidance influential range is $\eta_1 = 1$ m, and track-state influence range is $r_{\mathcal{S}} = L = 1$ m.

In the preliminary experiments, it was observed that in track state, the robots occasionally converged to a commanded speed that is too close to zero to be realized. Thus, in these experiments, a minimum velocity $V_{\min} = 50$ mm/s was imposed to prevent the robots from coming to a stop while attempting to reach equilibrium.

The first experiment presented utilizes a stationary rectangular spill with a perimeter length of 4.7 m. With five robots, and $r_{\mathcal{S}} = L = 1$ m, this should yield equal spacing along the perimeter once all robots achieve the track state. The results of this experiment are shown in Fig. 10.

From the figure, it can be seen that the first four robots quickly transition to track state. However, as the robots are not points but rather extended bodies, some additional time is required for the fifth robot to reach the perimeter. This is due to the avoidance hierarchy that forces a nontracking robot to yield to one that is tracking. Thus, a sufficient gap must be present before the robot can merge onto the perimeter. Once all five robots are on the perimeter, the speeds do indeed converge, and the spacing reaches an equilibrium.

The second experiment presented utilizes the same stationary spill of the first experiment, but with a second smaller spill with a perimeter length of 2.5 m that moves over time. The results of this experiment are shown in Fig. 11. Initially, the two spill components are overlapping; thus, the initial evolution is

[1] For these experiments, the robots do not make use of on-board cameras although this capability has since been implemented for these robots. The robots are now equipped with Logitech webcams and Hokuyo scanning laser rangefinders, and relatively localize to their neighbors using methods described in [31] and [32].

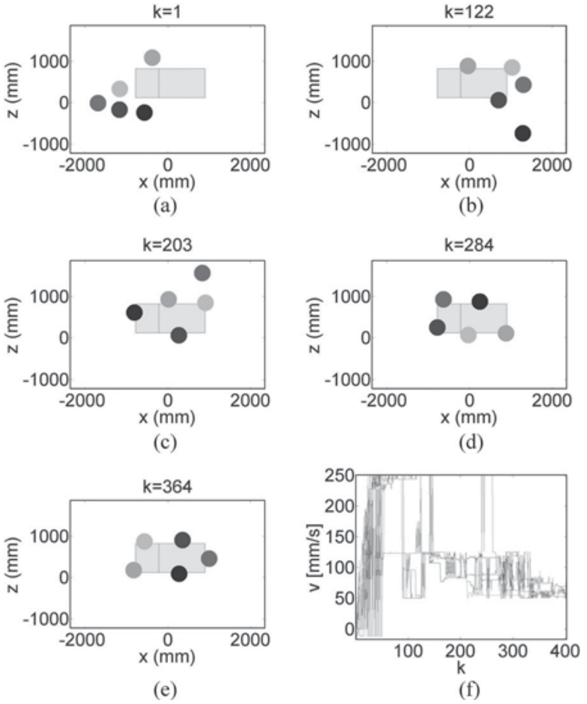


Fig. 10. Experimental result for five robots with a stationary large spill. (a)–(e) Snapshots of selected time points indicating the location of the spill and the agents. (f) Linear speeds of all agents.

similar to that of the first experiment. Beginning at time step 500, the smaller spill is gradually moved away from the larger spill, increasing the perimeter length and eventually separating into a second small spill; snapshots of this transition are seen in Fig. 11(d)–(e). During the growth phase of the spill, the robots dynamically increase their spacing to accommodate the longer perimeter. Once the smaller spill separates, one robot continues to track it with no change in its velocity; on the larger spill, the remaining four adjust their relative spacing to accommodate the shorter perimeter and lost robot.

At time step 700, the smaller spill rejoins the larger spill. This increased perimeter length, as illustrated in Fig. 11(f), is detected by the next-passing robot around time step 730. This causes a large increase in its speed because its leading robot is suddenly much further away along the perimeter. As seen in Fig. 11(h), the speeds of most of the robots briefly increase (around $k = 700$) when the smaller spill rejoins the larger spill. The reintroduction of the fifth robot into the network causes a disruption to the equilibrium, which quickly settles out. Finally, the small spill is completely engulfed by the larger spill, again resulting the convergence as seen in the first experiment.

Videos of the presented experiments and simulations, as well as other experiments not included here, can be found at the Duke RAMA Lab YouTube channel [33].

V. Conclusion

In this paper, a hybrid system of finite states with feedback control is advanced for a group of agents with a limited sensor FOV and a limited communication range to search for, detect,

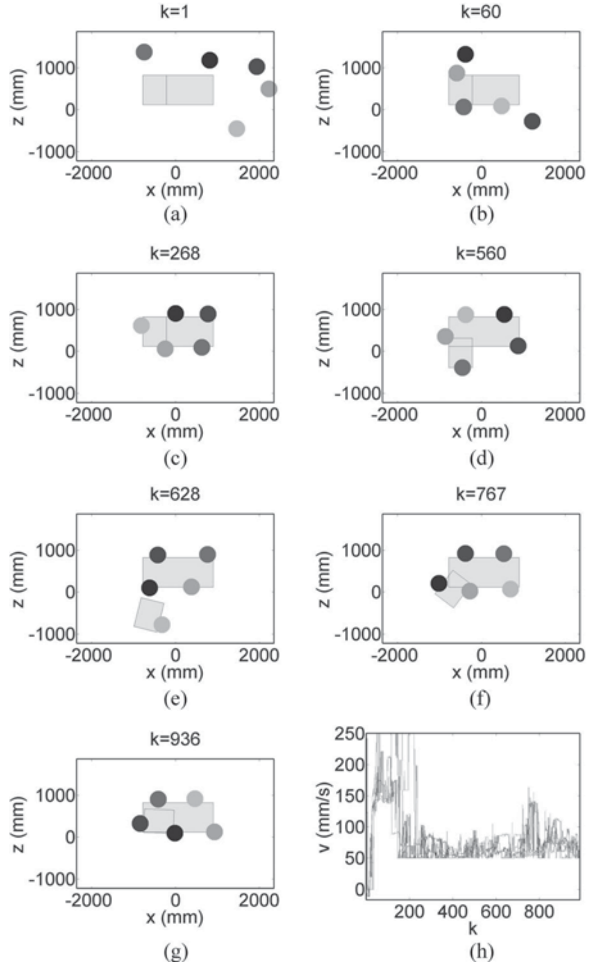


Fig. 11. Experimental result for five robots with a large spill that grows, splits into two spills, and then collapses to one large spill. (a)–(g) Snapshots of selected time points indicating the location of the spill and the agents. (h) Linear speeds of all agents.

and track a number of spills in an unknown environment. A hierarchical potential field is designed for agents in different states to control their movement that provides a simpler control law for collision avoidance. Simulation and experiment results demonstrate that the agents can successfully detect and track spills whose location, shape, size, and number may change over time. Significantly, this adaptation occurs automatically with no need to specify a predetermined distance between adjacent agents. Further focus on search efficiency and optimal convergence rates is ongoing. Additional enhancement of the robots' sensing ability continues.

Acknowledgment

The authors gratefully acknowledge the laboratory support provided by A. W. Caccavale and S. W. Li.

References

[1] E. Sahin, Ed., *Swarm Robotics* (Lecture Notes in Computer Science, vol. 3342). Heidelberg, Germany: Springer, 2005, ch. 2, pp. 10–20.

[2] T. H. Labella and M. Dorigo, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 1, pp. 4–25, Sep. 2006.

[3] G. Zhang and D. Garg, "Mobile multi-robot control in target search and retrieval," in *Proc. 1st Dyn. Syst. Control Conf.*, Ann Arbor, MI, 2008, pp. 677–683.

[4] F. Bourgault, A. Goktogan, T. Furukawa, and H. F. Durrant-Whyte, "Coordinated search for a lost target in a Bayesian world," *Adv. Robot.*, vol. 18, no. 10, pp. 979–1000, Dec. 2004, 2013.

[5] B. Shucker and J. K. Bennett, "Target tracking with distributed robotic macrosensors," in *Proc. MILCOM 2005*, Atlantic City, NJ, pp. 2617–2623.

[6] C. Schumacher, P. Chandler, and S. Rasmussen, "Task allocation for wide area search munitions via iterative network flow," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Monterey, CA, 2002, pp. 2002–4586.

[7] M. Alanyali, S. Venkatesh, O. Savas, and S. Aeron, "Distributed Bayesian hypothesis testing in sensor networks," in *Proc. Amer. Control Conf.*, 2004, vol. 6, pp. 5369–5374.

[8] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 163–175, Apr. 2009.

[9] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada, "Aseba: A modular architecture for event-based control of complex robots," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 321–329, Apr. 2011.

[10] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press, 1999.

[11] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization. An overview," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.

[12] D. Casbeer, D. Kingston, R. Beard, T. Mclain, S. Li, and R. Mehra, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *Int. J. Syst. Sci.*, vol. 37, no. 6, pp. 351–360, 2006.

[13] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA Trans.*, vol. 46, pp. 3–13, 2007.

[14] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control—A multivehicle platform for research in networked embedded systems," *IEEE Control Syst. Mag.*, vol. 27, no. 3, pp. 58–78, Jun. 2007.

[15] R. Alur, T. Dang, J. M. Esposito, R. B. Fierro, Y. Hur, F. Ivancic, V. Kumar, I. Lee, P. Mishra, G. J. Pappas, and O. Sokolsky, "Hierarchical hybrid modeling of embedded systems," in *Proc. First Int. Works., Embedded Softw.*, Tahoe City, CA, 2001, pp. 14–31.

[16] D. Bruemmer, D. Dudenhoeffer, M. Anderson, and M. McKay, "A robotic swarm for spill finding and perimeter formation," in *Proc. ANS Spectrum.*, Aug. 2002.

[17] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Contr.*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[18] R. Olfati-Saber and R. Murray, "Flocking with obstacle avoidance: Cooperation with limited communication in mobile networks," in *Proc. 42nd IEEE Conf. Decis. Control*, Maui, HI, 2003, pp., 2022–2028.

[19] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part I: Fixed topology," in *Proc. 42nd IEEE Conf. Decis. Control*, Maui, HI, 2003, pp., 2010–2015.

[20] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part II: Dynamic topology," in *Proc. 42nd IEEE Conf. Decis. Control*, Maui, HI, 2003, pp., 2016–2021.

[21] Z. Yang, Q. Zhang, and Z. Chen, "Choosing good distance metrics and local planners for probabilistic roadmap methods," *IEEE Trans. Robot. Autom.*, vol. 16, no. 4, pp. 442–447, Aug. 2000.

[22] G. Fricke, G. Zhang, A. Caccavale, W. Li, and D. P. Garg, "An intelligent sensing network of distributed swarming agents for perimeter detection and surveillance," in *Proc. 3rd ASME Dyn. Syst. Control Conf.*, Cambridge, MA, Sep. 2010, pp. 741–748.

[23] J. C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[24] A. Viguria and A. Howard, "An integrated approach for achieving multirobot task formations," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 176–186, Apr. 2009.

[25] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli, "A fast onboard relative positioning module for multirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 151–162, Apr. 2009.

[26] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Int. Conf. Decis. Control*, 2001, pp., 2968–2973.

[27] M. Kumar, D. Garg, and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," *IEEE Trans. Automat. Contr.*, vol. 55, no. 3, pp. 743–748, Mar. 2010.

[28] U. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

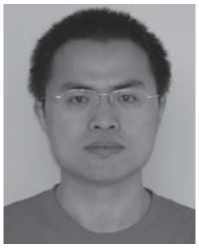
[29] G. Zhang, D. Garg, and G. Fricke, "Hazardous spill perimeter detection and monitoring via multiple autonomous mobile robotic agents," in *Proc. 3rd ASME Dyn. Syst. Control Conf.*, Cambridge, MA, 2010, pp. 639–646.

[30] G. K. Fricke, D. Milutinović, and D. P. Garg, "Sensing and estimation on an experimental testbed for swarm robotics," in *Proc. 2nd ASME Dyn. Syst. Controls Conf.,* Hollywood, CA, Oct. 2009, pp. 195–202.

[31] G. K. Fricke and D. P. Garg, "Discrimination and tracking of individual agents in a swarm of robots," in *Proc. 2010 Amer. Control Conf.,* Baltimore, MD, Jun. 30–Jul. 2, 2010, pp. 2742–2747.

[32] J. Huang *et al.*, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 205–215, Apr. 2006.

[33] J. Huang, S. Farritor, A. Qadi, and S. Goddard, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 205–215, Apr. 2006.



Guoxian Zhang received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 2003 and 2006, respectively, and the Ph.D. degree from Duke University, Durham, NC, in 2010.

He is currently a Software Quality Engineer at Microstrategy, Inc., Tysons Corner, VA. His current research interests include sensor planning, data mining, multiagent systems, Bayesian statistics, and decision making under uncertainty.

Dr. Zhang has been a member of the ASME since 2008.

Gregory K. Fricke received the B.Sc. degree in engineering and applied science with mechanical concentration from the California Institute of Technology, Pasadena, in 2000, and the M.Sc. degree in mechanical engineering in 2009 from Duke University, Durham, NC, where he is currently working toward the Ph.D. degree in mechanical engineering, focused on the control of multirobot systems using minimalist behavioral control laws.

His current research interests include system engineering, controls, and embedded software with several years of experience with The Boeing Company and Hughes Space & Communication.

Mr. Fricke is a student member of the ASME Dynamic Systems and Controls Division.

Devendra P. Garg received the B.Sc. degree in mechanical engineering from Agra University, Agra, India, in 1954, the B.Eng. degree from the University of Roorkee, Roorkee, India, in 1957, the M.Sc. degree in mechanical engineering from the University of Wisconsin, Madison, in 1960, and the Ph.D. degree in mechanical engineering from New York University, New York City, NY, in 1969.

He is Professor of Mechanical Engineering and Director of the Robotics and Manufacturing Automation (RAMA) Laboratory at Duke University, Durham, NC. He previously taught at the University of Roorkee (now, IIT/Roorkee), New York University, and MIT. He also served as Director of the Dynamic Systems and Control Program at the National Science Foundation for six years. He was a Guest Editor of two Special Issues of *ASME Transactions*. He is an author of two books and numerous research publications in technical journals in the U.S. and abroad. His research interests include control system synthesis, vehicle dynamics, robotics and automated manufacturing, and application of control theory to socioeconomic systems.

Prof. Garg is a Life Fellow of the American Society of Mechanical Engineers (ASME). He is a Past Chairman of the ASME's Dynamic Systems and Control Division (DSCD), and has served as Chairman of the Advisory Panel, and Chairman of the Honors and Awards Committee of the DSCD. He has received the ASME's Dedicated Service Award, the DSCD Leadership Award, and the 2003 Edwin F. Church Medal.

A.1.4 Modeling, Simulation, and Characterization of Distributed Multi-agent Systems

The following paper (on the next 8 pages) was co-authored by Reed F. Young and Devendra P. Garg, and appeared in 2012 in Volume 10, Number 2 of the Journal of Systemic, Cybernetics and Informatics on pages 73–80.

Modeling, Simulation, and Characterization of Distributed Multi-agent Systems

Reed F. YOUNG
U.S. Army Yuma Proving Ground
Yuma, AZ 85365

and

Devendra P. GARG
Mechanical Engineering and Materials Science Department, Duke University
Durham, NC 27708 USA

ABSTRACT

A strategy is described that utilizes a novel application of a potential-force function that includes the tuning of coefficients to control mobile robots orchestrated as a distributed multi-agent system. Control system parameters are manipulated methodically via simulation and hardware experimentation to gain a better understanding of their impact upon mission performance of the multi-agent system as applied to a pre-determined task of area exploration and mapping. Also included are descriptions of experiment infrastructure components that afford convenient solutions to research challenges. These consist of a surrogate localization (position and orientation) function utilizing a novel MATLAB executable (MEX) function and a user datagram protocol (UDP)-based communications protocol that facilitates communication among network-based control computers.

Keywords: Control, Multi-agent System, Swarming Robots, Potential Function.

1. INTRODUCTION

From the inception of the development of robotic technologies, researchers have generally concluded that, with many if not most applications, there are significant advantages to be gained through the coordination and orchestration of multiple robotic entities [1-2]. This is not difficult at all to imagine by simply observing Darwinian development in biological systems whereby two or four legs proved superior to one for locomotion; or separate arms each with a hand, fingers, and an opposing thumb proved superior for grasping and cooperatively manipulating objects. Furthermore, it similarly is understandable that a small group of separate robotic entities (e.g., a small number of cooperating or coordinating unmanned ground vehicles) could also exhibit significantly desirable characteristics in certain circumstances. Five main motivations for using multi-robot systems include: task complexity; task distribution; resource distribution; parallel processing; and robustness through redundancy [3]. Some researchers have explored the advantages gained for specific tasks such as exploration where a specified task can be accomplished by a number of agents surveying the maneuver space versus a single robot tasked with exploring the entirety by itself [4]. Indeed, these become valuable against overarching task accomplishment; however, not without burden. Multi-robot systems add communications overhead and workload with the requirement for inter-robot cooperation and knowledge. They also add complexity to path planning and collision avoidance schemes since each robot's cooperating peer also exists as an obstacle to be avoided [5].

Research over the past years has centered on organizing several to tens of mobile robots (or agents) into coordinated multi-agent systems (MAS) [6-8]. Further, the quantity of these systems may reach into the hundreds or thousands that comprise a robotic swarm. In contrast to single mobile robots with their broad, comprehensive capabilities (vision, LIDAR, differential global positioning, ultrasonic proximity sensing, etc.), the agents comprising a MAS tend to have somewhat lesser organic capability due to a desired affordability or individual physical size. An example of one such capability is localization that could provide accurate agent position and orientation in a global reference akin to the Global Positioning System (GPS). While GPS receivers have become small and inexpensive, they cannot be used indoors where MAS experimentation and utilization typically exist. Research in the field has centered on understanding the root mechanisms and control features that can define and drive the behavior and capability of the MAS architectures in terms such as task allocation, flocking, or foraging. The study of analogous biological systems has also provided valuable insight. However, to date, there has been little work done with respect to understanding the capability of the MAS as a function of its composition given the option of pre-mission selection of the performance and capability of individual agents, as well as opportunity to include extra-system information sources or sensors.

Various attempts at defining MAS composition [9-11] explored heterogeneous versus homogeneous systems, highlighting concepts such as hierarchic social entropy, diversity, and cooperative localization. Heterogeneous MAS have been used to explore unknown environments for the purpose of mapping and exploration [12]. Here, each MAS featured varying types of single-mode sensor payloads including, for example, an infrared camera or sonar, coordinated in an occupancy grid Bayesian mapping algorithm. But again, the composition of the architecture was held constant.

An example of a MAS configuration is illustrated in Figure 1. Here, four differential-wheeled robotic agents featuring distributed control architectures orchestrate to conduct a mapping of an initially-unknown discrete arena to localize and identify targets and obstacles. Each agent has imaging and proximity sensing. They pass information to each other via a consolidated state matrix that receives data from each agent and simultaneously provides state updates to each agent.

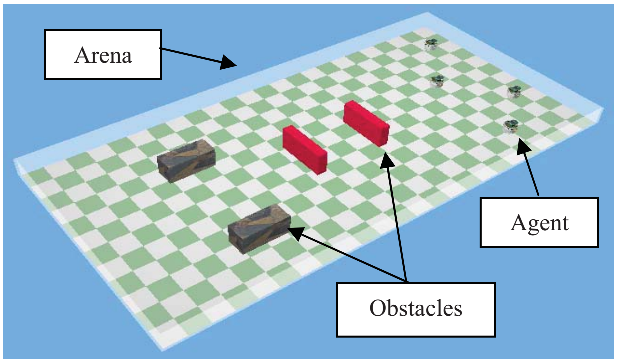


Figure 1: Representative Multi-agent System Arena.

The underlying control architecture plays an absolutely critical role in the efficiencies observed in the multi-agent versus single-robot architectures [13-16]. Clearly, beyond the inherent capability of the electro-mechanical configuration as well as the sensor payloads, the single greatest contributor to mission efficiencies and task capabilities will exist in the capability, adaptability, and comprehensiveness of the control system. A wide variety of algorithms and schemes have been explored including feedback laws, leader follower, task-oriented mission planning, task reallocation and reordering, and deliberative and reactive behavior modeling. Beyond these, potential-function-based robot control strategies exhibit significant promise to control payload-agile MAS.

This paper is organized as follows: Section 2 presents a potential-function based robot control strategy that uses parametric tuning factors to increase the efficiency with which the control strategy can accomplish an exploration task with no a priori information of the search space. Section 3 describes results from the research's simulation phase. Section 4 describes the creation of a novel surrogate global positioning system based on a task-specific MATLAB MEX function sampling an optical motion capture system broadcasting over a network socket. Additionally, this section summarizes a UDP-based data transmission architecture that enabled inter-agent sharing of realized occupancy map elements. Sections 5 offer the results from the research's experimentation phase whereby increasing quantities of agents with various capabilities were placed in the arena and evaluated for task accomplishment metrics such as time to complete. The last sections summarize the research results and conclusions.

2. POTENTIAL-FUNCTION-BASED ROBOT CONTROL STRATEGY

Control via potential-function theory has proven to be a valuable basis upon which more complex control feedback influence can be implemented. In particular, potential-function theory affords a comprehensive and simple method to add factors that can be scaled either as independent input functions or those dependent upon selected system features. The basis of the potential function is to represent system control influences as independent or dependent forces [17] akin to an electrical potential field. Each force is calculated based on physical parameters (such as real distances) or on representative values (such as arbitrary repulsion to facilitate obstacle avoidance or attraction to a desired target location). In conjunction with the potential function, it is convenient to represent the operational space as an occupancy map, i.e., a two- or three-dimensional grid that sub-divides the space into identifiable cells. Each cell would have characteristics such as a potential related to its occupancy status, or status as a target destination. Since the potential function depends upon proximity, the relative location of the cells would also be cataloged.

The goal of the control function for the MAS is to calculate the agent-unique potential force witnessed by each of the agents comprising the MAS. Force defines the resultant heading and speed that the individual agent would follow. The potential force existent on the ith agent is given by:

$$\hat{F}_i = Q_i\left(\sum_{j=1}^{N_C}\frac{\mu_1 q_j}{\left\|rad_j\right\|^2}\widehat{rad_j} + \sum_{j=1}^{N_A-1}\frac{\mu_2 Q_j}{\left\|rad_j\right\|^2}\widehat{rad_j}\right) + \sum_{j=1}^{L}\mu_3 P_j \qquad (1)$$

where,

Q_i = inherent charge of the ith agent,
Q_j = inherent charge of the jth agent,
q_j = inherent charge of the jth cell,
rad_j = radius vector from the ith agent to the jth agent or jth cell,
N_C = number of cells in the occupancy map,
N_A= total number of agents,
P = calculated force contribution from specified payloads, and
μ_x = tuning coefficients.

Then, the resultant heading angle (θ_i) of the ith agent is given by:

$$\theta_i = \tan^{-1}\left(\frac{F_{Z\,i}}{F_{X\,i}}\right) \qquad (2)$$

The calculation of P will vary with payload type as the value and weighting of each payload type can and should have varying degrees of influence on the resultant potential force calculation.

Tuning coefficients (μ_x) provide a mechanism whereby the relative contribution of each of the potential forces can be directly tuned either as an independent variable, or dynamically tuned based on a feedback mechanism. This feature is particularly important in multi-agent systems due to scalability. Since the potential forces contributed by elements within the system are cumulative, it becomes very important to gain an awareness of the proportional contributions and to subsequently be able to control them proactively. To illustrate, assume an occupancy grid of 120 by 240 cells or a total of 28,800 total cells. Each cell might offer a contributing charge value from 0 to 1 meaning a span of 28,800 charge units. If other contributors merely summed to a maximum of 1,000 charge units, the cells' contribution would easily overwhelm any potential contribution calculation.

The occupancy map cell characteristic vector represents the probability that the cell is occupied. A value of 1.0 indicates that there is a 100 percent chance that the cell is occupied. A value of 0.0 indicates that there is a 100 percent chance the cell is unoccupied. A value of 0.5 would indicate an equal probability that the cell is either occupied or unoccupied. Separate variables that provide valuable influence in the force calculation are the number of times that a cell has been sampled by a sensor, referred herein as the "cell visit count," and the number of times that a fraction of the entire occupancy map has been visited. The usefulness here is to accommodate the dynamic nature of agents moving in the arena as well as the potential for moving targets in the field of regard. As a result, the charge associated with each cell is given by:

$$q_i = \begin{cases} P_O q_{max} & \text{for } 0<V<n \\ 0 & \text{for } V>n \end{cases} \qquad (3)$$

where,

P_O = cell occupancy probability,

q_{max} = maximum possible cell charge,

V = cell visit count defined as the number of visits to the i^th cell, and

n = number of times that a percentage of the cells have been visited.

3. INITIAL OBSERVATIONS THROUGH SIMULATION

Initially, the Cyberbotics' Webots simulation package was used to represent a 120 by 240 centimeter arena (Figure 1) occupied by e-puck differential-wheeled robots. The arena is subdivided into a representative occupancy mapping grid consisting of 1 cm by 1 cm cells each having an associated vector representing dynamic characteristics such as the probability of occupancy or potential charge. The MATLAB scripting language was utilized to provide high-level potential-function based path planning as well as map databasing.

A comprehensive set of state and evaluation metrics are collected at every control time step that provide an opportunity to observe the progression of the exploration task. Agent trajectories (Figure 2) provide insight to the energy consumed by each individual agent in terms of the distance travelled. Trajectories also illustrate the qualitative uniformity and dispersion with which the team of agents uniquely maneuver in and canvas the arena.

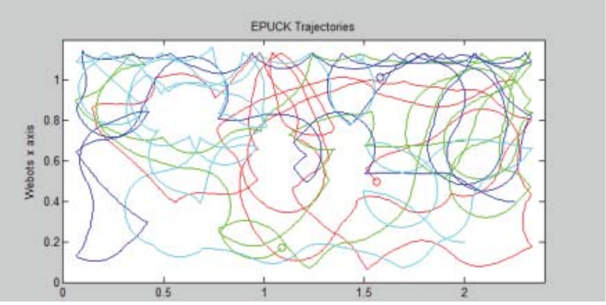


Figure 2: Agent Trajectories.

The occupancy map data vector (Figure 3) at a specific time step is displayed to show the occupancy probability, cell charge, and cell visit count. Observed individually and sequentially over time, these vectors provide insight to the progression of understanding of the heretofore unknown arena and resolution with which obstacles and targets appear. One can also conclude trends such as the extent of exploration and behaviors proximate to obstacles.

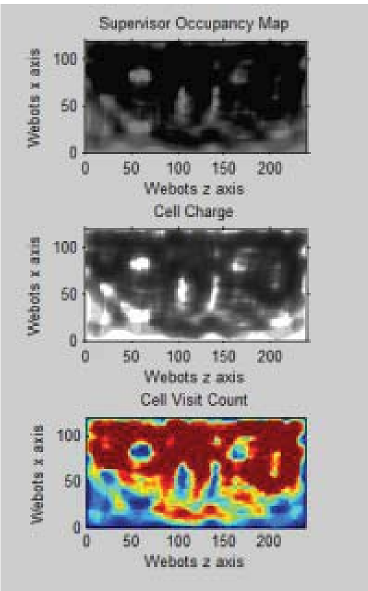


Figure 3: Occupancy Map Data Vectors.

Finally, mean characteristic values graphed over time (Figure 4) quantify the performance of the multi-agent system and illustrate its ability to progress toward threshold goals. The average occupancy map value shows the time progression of the average across each cell's probability of occupancy. Since there is no a priori knowledge afforded, the initial value is 0.5 meaning all cells have equal probability of being occupied or unoccupied. Over time, as the agents explore the arena and determine free and occupied space, the curve progresses as expected decreasing over time with the free space determined and step increasing periodically with the sensor-enabled detection of obstacles, i.e., occupied space. The curve is non-linear due to the proximity of unexplored space diminishing over time resulting in the necessity to travel longer distances (and consuming more time) to sample those unexplored spaces. The average cell charge plots the average of all the cells' calculated charge values over time. This curve initially decreases due to the fact that as cells are explored, their charge decreases so as to have less impact in subsequent path planning loops. However, in deference to the dynamic state of the arena, the average cell charge begins to increase in correlation with the map coverage count due to the forcing relationship in Equation (3). The average cell visit count tracks the average number of times that any agent samples individual cells over time. This curve has a constant slope as expected knowing that the agents' speed varies very little perhaps only to slow for short period of times while maneuvering around obstacles. Finally, the map coverage count (n, Equation 3) is an evolving parameter that tracks the number of times that a pre-defined percentage of the arena has been sampled at least the count's number of times. For example, the count starts at zero. As soon as 75 percent of the arena has been sampled once, the count increments to 1. As soon as 75 percent of the arena has been sampled twice, the count increments to 2. This parameter is used within the control loop to influence the charge calculation in order to accommodate the dynamic nature of the state representation. The shape of this curve provides qualitative clues as to the efficiency of the search.

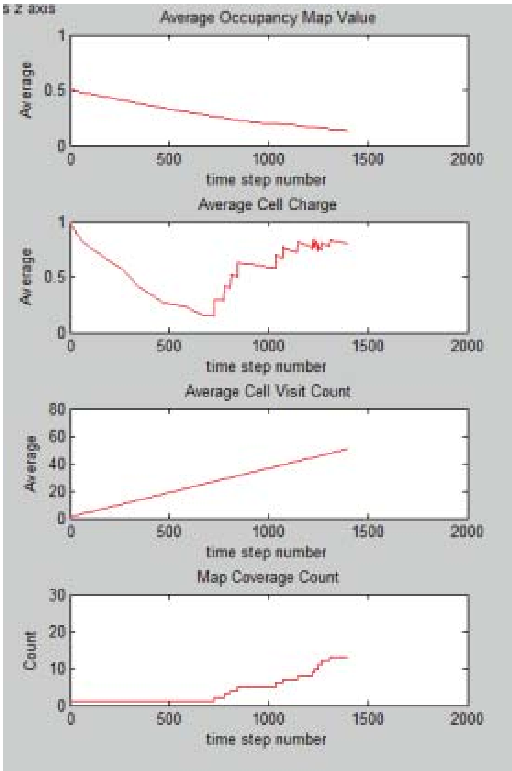


Figure 4: Characteristic Value Histories.

4. DISTRIBUTED MAS EXPERIMENTATION

The transition from simulation to hardware experimentation is a non-trivial task. There are many real-world challenges in localization and communications that can be inaccurately represented as perfection in the simulation. In support of on-going MAS research at Duke University, an arena (Figure 5) has been created consisting of a 120 by 240 centimeter walled stage hosting various numbers of e-puck robots [18], obstacles, and targets. The e-puck robot is approximately 7 cm in diameter. It features a Microchip dsPIC processor running at 60MHz or about 15 MIPS. Mobility is provided by two stepper motors in a differential-wheel drive configuration. The robot has eight infra-red sensors positioned around the robot's perimeter that provide both proximity sensing and light sensing. There is a 640x480 pixel camera on-board that can provide video and imagery in either color or black and white modes. Bandwidth limitations force pragmatic collection to 52x39 or 640x1 pixel images at up to three frames per second. There are three omni-directional microphones for sound localization, a three-axis accelerometer, and on-board speaker. Communications is provided by Bluetooth. Each robot has a dedicated desktop computer that remotely controls the robot via Bluetooth communications. This dedicated computer allows for substantial processing capability, image processing in particular, that simply isn't supportable via the e-puck limited on-board processing capability. The control software is written in MATLAB programming language. The e-puck robots have no inherent localization capability.

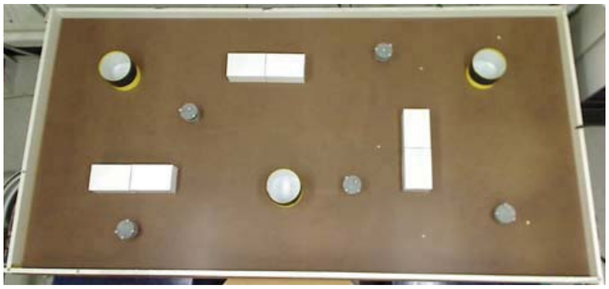


Figure 5: MAS Experimental Arena.

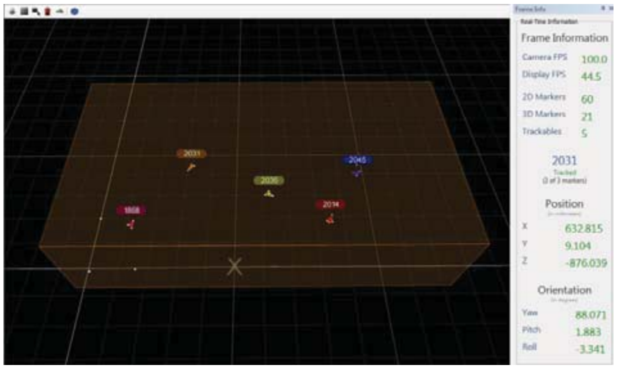
Two novel components of this experimentation infrastructure required to fully replicate the simulation environment include a surrogate global positioning system (GPS) and data packet transfer between agent controllers. Both capabilities, developed under the auspices of this research, are described in the following subparagraphs.

Surrogate GPS via External Optical Motion Capture and Tracking

Much research effort thus far has been focused on the simultaneous localization and mapping (SLAM) problem [19]. SLAM acknowledges that externally-provided localization information is not typically available, and also acknowledges a scenario of exploration of heretofore unknown space. Therefore, clever schemes are explored that use available modalities (e.g., inter-agent communications) to determine relative localization among the agents. That, combined with robot-sensor-enabled, continually-updated mapping databases, generates a comprehensive representation of the spatial and spectral environment. Many times though, inherent SLAM functionality is not germane to the research problem being pursued. In this case, an extra-MAS localization capability is most desirable to minimize the experiment's independent variables.

There are several materiel possibilities for an extra-agent surrogate GPS capability. Fricke et al [20] utilized a combination of a Cognex camera and Sick LIDARs to calculate localization. The Vicon motion capture system [21] is a popular solution that uses connected high-speed digital cameras viewing the MAS experiment workspace with redundant coverage to provide localization of tracked objects.

The Natural Point OptiTrack™ system [22] utilizing its 'Tracking Tools' (TT) software application (Figure 6) provides a similar capability. The OptiTrack™ cameras are installed and calibrated for the desired experiment workspace. Each robot is given a unique spatial configuration of reflective markers (Figure 7) that enables individual robot identification during the tracking process. A critical functionality important to supported experimentation that TT offers is the ability to multicast tracked-object localization in real time through a designated network port on the computer hosting the Tracking Tools application. If connected to a local-area network, any other computer (e.g., one hosting an agent controller) could also have near-instantaneous access to the real time streaming position information of itself, and of any other agent in the workspace with update rates approaching 100 Hertz. This is particularly desirable as this streaming broadcast scheme significantly reduces the inter-agent communications requirement. TT offers three formats for broadcast: industry-standard Virtual Reality Peripheral Network (VRPN) and Trackd®, and NaturalPoint's own NatNet format. VRPN, provided by the University of North Carolina's Department of Computer Science, offers several advantages including a comprehensive C/C++ dynamic-linked library (DLL) and public domain license [23]. For illustration, NaturalPoint provides sample C++ code titled "VRPN-Listener" [24] that affords a simple capability for sampling to the TT port and output the received localization information to the screen.



Frame Information

Camera FPS 100.0
Display FPS 44.5
2D Markers 60
3D Markers 21
Trackables 5

2031
Tracked
(3 of 3 markers)

Position
X 632.815
Y 9.104
Z -876.039

Orientation
Yaw 88.071
Pitch 1.883
Roll -3.341

Figure 6: Screen Shot of NaturalPoint Tracking Tools in Use.

Control can be coded in a wide variety of software languages, each offering unique benefits and shortcomings. MATLAB offers significant advantages in terms of comprehensive image processing routines and robust matrix manipulation routines to support important functionality such as volume mapping. In implementation with the VRPN construct though, a significant challenge arises in that there are no MATLAB native commands or functions available to interpret the VRPN broadcast as the DLL is C/C++ based. Here we have realized a solution and its experimental implementation by creating a MATLAB-based MEX-function [25] that allows the robots' MATLAB-based control algorithms to access and call the C++ DLL native to VRPN and thereby provide a comprehensive surrogate localization function.

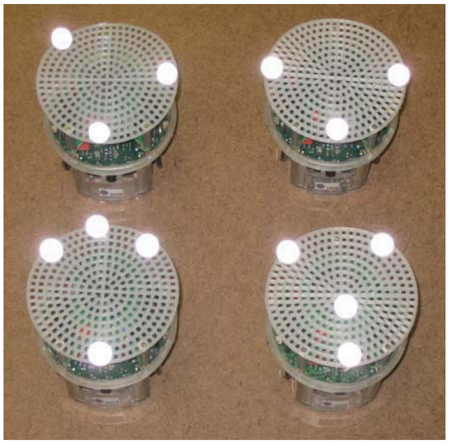


Figure 7: Optical Tracking Reflectors on E-puck Robots.

MEX-functions have an organic capability offered by MATLAB that allows a programmer to compile software routines written in C/C++ and linked to C/C++ DLLs resulting in a function callable from within MATLAB programs just as one would call a MATLAB built-in function. The specific C++ code used in this research is a task-specific modification of the Natural Point-provided "VRPN-Listener" code written in C++. The capability of the created MEX-function is twofold. First, given identification of the specific agent of interest, the MEX-function interrogates the VRPN broadcast port and obtains the most current localization information. Secondly, the MEX-function converts the provided localization information into the desired coordinate system and units. For the cited experiment, this consisted of converting the VRPN-provided orientation in the three-dimensional quaternion rotation space in radians to roll-pitch-yaw parameters in degrees.

UDP Communications for Data Sharing

Multi-agent systems can be characterized as having either distributed or centralized control with the distinction being the level to which each individual agent in the system governs its own motion. Fully distributed control is characterized by a complete absence of data or control-command exchange among the agents whereby each agent relies upon its own sensors to understand the environment, and its own processing capability to determine controlled actions. On the other end of the spectrum, centralized control is characterized by a single processor calculating the collective motion of the system and disseminating commands to each agent for the orchestration movement. As a result, distributed control architectures require robust on-agent processing capabilities, but minimal communications. In contrast, centralized control architectures are accomplished with a single robust processor (the central controller) and relatively minimal agent processing. However, a robust inter-agent communication capability is required to facilitate distribution of necessary control commands, sensor information, and other data such as state and localization information. As described control is a spectrum, one can imagine that there are varying levels of processing and communications required among the various control schemes ranging from fully distributed to fully autonomous.

The agents comprising the system in this research are constructed and defined as being fully distributed with each agent and its associated controlling computer fully responsible for calculating motion commands. However, there is a slight variation in that a separate "state" computer exists to act as a surrogate global positioning system (via the OptiTrack™ Tracking Tools system and software); and to collect, maintain, update, and distribute state information (position and orientation of each agent as well as occupancy mapping of the operating space). Because of this, a comprehensive inter-computer communications infrastructure and protocol was created for passing of data among the state computer and controller computers.

From a hardware perspective, the seven computers (one state computer plus the six controller computers) achieve full gigabit Ethernet inter-computer communications throughput capability by connection to a single DLink DGS-2208 eight-port gigabit Ethernet switch. That switch in turn, is connected to a Netgear Prosafe FVS336G gigabit Ethernet router. That router provides connectivity to the external internet service provider, but more importantly, is set up to statically assign internet-protocol addresses to each of the experiment's computers. With this architecture, each computer can be uniquely identified by its statically-assigned IP address facilitating unique computer programming.

The second aspect of providing a robust information exchange infrastructure is to select the underlying data protocol used. There are several protocols available that can be used in MATLAB and C programming including notably transmission control protocol/internet protocol (TCP/IP) and user datagram protocol (UDP). In fact, MATLAB's Instrument Control Toolbox provides a very comprehensive set of TCP/IP and UDP command sets. Each offers its own advantages and disadvantages. For example, TCP/IP offers a strong recovery feature that consists of error-checking that will initiate the resending of data packets if they are not accurately received. UDP offers greater throughput potential due to its use of minimal data fields for error-checking overhead. However, it can suffer some packet losses. Ultimately, UDP was utilized based on the construct of the validation experiment. In fact, MATLAB's TCP/IP protocol cannot be used to communicate between disparate MATLAB instantiations, i.e., communications between MATLAB running on computer "X" and a separate instantiation of MATLAB running on computer "Y." The concern here, however, is that while it offers the greatest throughput, UDP does not offer any organic error checking. This turned out not to be an issue in this research as error checking in the authored code itself was able to accommodate any transmission errors.

5. RESULTS

Simulation

Digital simulations showed that the potential-function path planning control architecture provides a robust and complete methodology to explore and map an initially unknown operating space. One representative experiment included a series of excursions in which the time allowed per mission was held constant. The number of agents comprising the MAS was varied from one to four. These variant MAS were then evaluated in terms of the resulting state of knowledge such as map coverage count or average occupancy probability. Increases in number of agents utilized in sequential experiments consistently showed a non-linear relationship between number of agents and perforamance criteria, for example, map coverage count (Figure 8). This likely is due to an inherent efficiency gained in terms of distance required to travel to unexplored spaces by having additional agents in the arena.

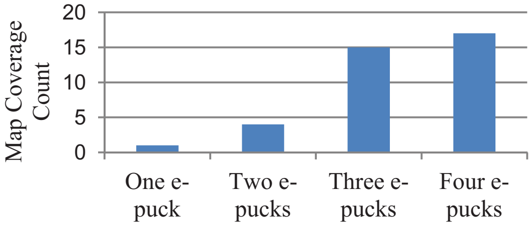


Figure 8: Map Coverage Count versus Number of Agents.

The physics of potential energy functions dictate that the force is inversely proportional to the square of the distance. In this research, the energy physics were replicated as such. However, Figure 9 shows that this tends to greatly magnify the influence of proximate cells and greatly diminish distant cells. As a result, calculated path planning was heavily weighted in a more local versus global sense. This clearly will influence the efficiency of the MAS' ability to exhaustively search the entire arena acknowledging that understanding the entirety of unexplored space and accommodating in the path planning scheme is likely more effective than merely considering the radius to unexplored space. Nonetheless, the shortcoming of this inefficiency is overwhelmed by the potential function's ability to accomplish exhaustive search. Ultimately, each cell will influence the path planning calculation insuring that each cell ultimately is explored. An enhancement to consider is to normalize cell charges to and gain better responsiveness over the entirety of the stage versus proximate cells. Effectively, this means breaking the laws of physics and making the cell charge vary as reduced exponents or even linearly as a function of distance. Clearly, this deviates from classic potential theory, but might nonetheless offer some interesting advantages in path planning efficiencies.

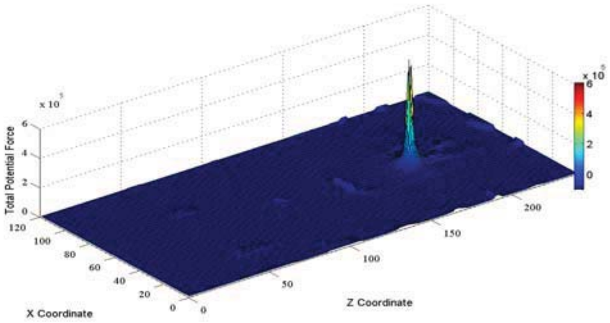


Figure 9: Total Potential Force Witnessed by an Individual Agent in the Arena.

Experimental Validation

The novel surrogate-localization and inter-agent data-communications solutions developed for experimentation infrastructure proved to be robust and highly conducive to support control algorithm cycle rates. Based on the data collection capability and mobility of the individual agents, governing control schemes strive for cycle times less than 200 milliseconds to insure exhibition of responsive behaviors. The surrogate localization function created only minimal processing time burden consuming only low-single-digit milliseconds and with near-real-time latencies. This proved extremely supportive of the control scheme with the agents being able to accurately represent the locations of themselves, other agents, targets, and obstacles - certainly well within the 1 cm^2 resolution of the state map. A significant enabler to this was an ability to calibrate the accuracy of the OptiTrack^{TM} system itself to sub-millimeter levels. Additionally, this function proved a robust capability to provide the variety of network-connected controllers immediate access to their own controlled agent, but also to the peer agents operating in the arena, all in near real time, and without having to burden the inter-agent communications architecture. The UDP-based data-communications function provided a reliable capability to pass large datasets of state information among the networked computers. A single state map consists of 28,800 cells (120 by 240) each represented by a 64 bit data element. Therefore, the single full mapping approaches 1.85 megabits of data. Over the course of an individual experiment, hundreds of state maps are transmitted to the agent controllers along with thousands of state map updates from the agent controllers. This means tens of thousands of packets navigating the network among the state and controller computers. Notwithstanding UDP's lack of an error correction capability, the data-communication solution herein resulted in zero packet loss realized across more than 300 individual experiments. Certainly, this was enabled by isolating the network on a single 8-port network switch; however, also enabled by careful consideration and definition of ports and buffers.

The resulting experiment infrastructure enabled a comprehensive series of individual vignettes and excursions that explored varying MAS parameters such as the number of imaging payloads, number of agents comprising the MAS, and relative agent and cell charge values. These were evaluated against mission performance criteria such as time to explore a selected percentage of the identified space or the number of positive detections collected by proximity sensing or imaging. With these parameter and metric sets, methodical experiments can be conducted sequentially to result in datasets that can be reduced and interpreted for qualitative and quantitative characterization of the relative and absolute performance of the various MAS configurations.

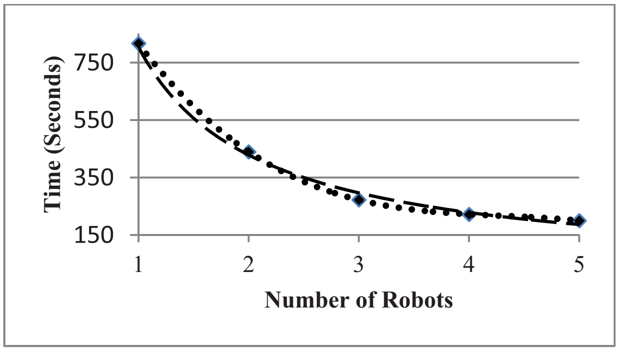


Figure 10: Time to Complete the Mission versus Number of Robots.

The most interesting conclusion drawn was that the relative performance of a MAS as a function of the number of agents comprising the MAS faded non-linearly (Figure 10). In these series of experiments, mission success was defined as exploring 75% of the arena's cells. Further, each agent was capable to use its on-board camera and associated image processing capability to detect and localize targets in the arena. Each agent could also use its proximity sensing for local collision avoidance and for obstacle detection and localization. Excursions were run whereby the number of agents comprising the MAS was varied from one to five with ten experiment runs per excursion. Mean values were taken across the ten runs to conclude the excursion results. In an attempt to curve fit the data (annotated by the diamonds), both polynomial and power types were evaluated. A third-order polynomial curve (dotted line) provided a better fit (R^2=1) within the bounds of one to five agents; however, did not extrapolate as values above six agents drove the time to completion to negative numbers. A power curve fit (dashed line) also resulted in an excellent fit (R^2=0.9898) and additionally provided better extrapolation beyond five robots with 6- through 8-agent MAS resulting in 158, 137, and 122 seconds, respectively. Additionally, this curve fit has an asymptote at 0 seconds as one would expect. Logically, this is the case whereby the entirety of the arena is covered by the concatenation of agents' sensors such that that entirety is exhaustively sensed in the very first control cycle's data collection.

6. CONCLUSION

A solution is described that utilizes a novel application of a potential-force function that includes tuning coefficients to control mobile robots orchestrated as a distributed multi-agent system. Further, MAS and individual agent parameters were methodically explored to evaluate and characterize resulting overarching performance of the MAS. Experimentation began via a software simulation. Then, with the creation of novel localization and data-transfer capabilities, the experimentation migrated to a hardware implementation. Results provide valuable insight to the behavior of the various MAS configurations such that a priori mission planning can be optimized as a function of prioritized and weighted criteria and planning factors. Notably, there is a distinct tradeoff between MAS capability and consumption such that optimal configuration can indeed be selected a priori to conduct of missions.

7. ACKNOWLEDGEMENT

The research reported in this paper was supported by the U.S. Army Research Office under grant number W911NF-08-0106.

8. REFERENCES

[1] D. Garg and R. Young, "Coordinated Control of Cooperating Robotic Manipulators", *International Journal of Systems Science*, Vol 21, Issue 11, November 1990 , pp. 2161 - 2176.

[2] R. Young, "The Coordination of Multiple Robotic Manipulators", Master's thesis, Duke University, 1987.

[3] L. Parker, "Multiple Mobile Robot Systems", *Springer Handbook of Robotics*, 2008, pp. 921-941.

[4] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated Multi-robot Exploration", *IEEE Transactions on Robotics*, Vol. 21, No. 3, June 3, 2005, pp. 376-386.

[5] K. Baghaei, and A. Agah, "Task Allocation Methodologies for Multi-Robot Systems", *Technical Report ITTC-FY2003-TR-20272-01*, November 2002.

[6] G. Fricke, G. Zhang, A. Caccavale, W. Li, and D. Garg, "An Intelligent Sensing Network of Distributed Swarming Agents for Perimeter Detection and Surveillance", Paper No. DSCC 2010-4256, *Proceedings of the ASME 2010 Dynamic Systems and Control Conference*, September 12-15, 2010, Cambridge, MA.

[7] D. Atta, B. Subudhi, and M. Gupta, "Cohesive Motion Control Algorithm for Formation of Multiple Autonomous Agents," *Journal of Robotics*, May 31, 2010, pp. 1-13.

[8] R. Williams and J. Wu, "Dynamic Obstacle Avoidance for an Omnidirectional Mobile Robot," *Journal of Robotics*, September 14, 2010, pp. 1-14.

[9] L. Parker, “Guest Editorial, Special Issue on Heterogeneous Multi-robot Systems”, *Autonomous Robots*, Vol 8, Issue 3, June 1, 2000, pp. 207-208.

[10] L. Parker, “The Effect of Heterogeneity in Teams Of 100+ Mobile Robots”, *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings of the 2003 International Workshop on Multi-robot Systems*, Vol II, March, 2003, pp. 205-218.

[11] M. Potter, L. Meeden, and A. Schultz, “Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergence of Specialists”, *International Joint Conference on Artificial Intelligence*, Vol 17, Part 1, 2001, pp. 1337-1343.

[12] R. Grabowski, L. Navarro-Serment, C. Paredis, and P. Khosla, “Heterogeneous Teams of Modular Robots for Mapping and Exploration”, *Autonomous Robotics*, Vol 8, Number 3, June, 2000, pp. 293-308.

[13] J. Desai, J. Ostrowski, and V. Kumar, “Controlling Formations of Multiple Mobile Robots”, *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, Vol 4, May 1998, pp. 2864-2869.

[14] R. Alur, A. Das, J. Esposito, R. Fierro, G. Grudic, Y. Hur, V. Kumar, J. Ostrowski, G. Pappas, B. Southall, J. Spletzer, and C. Taylor, “A Framework and Architecture for Multirobot Coordination”, *Experimental Robotics VII*, 2001, pp. 303-322.

[15] J. Kim, D. Shim, S. Rashid, and S. Sastry, “Hierarchical System for Multiple-agent Scenarios”, *DTIC Report A794314*, January 1, 2002.

[16] B. M. Dias, B. Browning, M. Veloso, and A. Stentz, “Dynamic Heterogeneous Robot Teams Engaged in Adversarial Tasks”, *Technical Report CMU-RI-TR-05-14*, Robotics Institute, Carnegie Mellon University, April, 2005.

[17] A. Scott, and C. Yu, “Cooperative Multi-agent Mapping and Exploration in Webots”, *Proceedings of the 4th International Conference on Autonomous Robots and Agents, Canberra, Australia,* Feb 2009, pp.56-61.

[18] e-puck Educational Robot. Available:www. e-puck.org.

[19] X. Hou, C. Yu, and T. Summers, “A Virtual Framework of Robotic SWARM Testbed”, *Proceedings of the 21st Annual International Conference on Chinese Control and Decision Conference*, June 17-19, 2009, pp. 930-935.

[20] G. Fricke, D. Milutinovic, D. Garg, "Sensing and Estimation on a Modular Testbed for Swarm Robotics", *Proceedings of the 2nd Annual Dynamic Systems and Control Conference*, October 12-14, 2009, Hollywood, CA, paper TuBT4.4.

[21] Vicon website. http://www.vicon.com.

[22] NaturalPoint website. http://www.naturalpoint.com.

[23] Virtual Reality Peripheral Network website. http://www.cs.unc.edu/Research/vrpn/.

[24] Natural Point’s VRPN Streaming Sample Distribution Files. Available: http://media.naturalpoint.com/software/OptiTrack_files/VRPN-Tracking-v3.zip.

[25] MathWorks MATLAB “MEX-file Guide.” Available: http://www.mathworks.com/support/tech-notes/1600/1605.html#example5.

A.2 Peer-reviewed Conference Proceedings

A.2.1 Self-sorting in a Swarm of Heterogeneous Agents

The following paper (on the next 6 pages) was co-authored by Manish Kumar, Devendra P. Garg, and Vijay Kumar, and appeared in the Proceedings of the 2008 American Control Conference on pages 117–122, published and presented in June 2008.

2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Self-sorting in a swarm of heterogeneous agents

Manish Kumar, Devendra P. Garg, and Vijay Kumar

Abstract— **Sorting of heterogeneous units is a self-organized behavior which is seen in many biological systems. One of the best examples of such systems is a system of biological cells of heterogeneous types that has the ability to self-organize into specific formations, form different types of organs and, ultimately, develop into a living organism. Earlier research in this area has indicated that such self-sorting behaviors in biological cells and tissues are made possible because of difference in the adhesivity between different types of cells or tissues. Inspired by this differential adhesivity model, this paper presents a decentralized approach based on differential artificial potential to achieve the self sorting behavior in a swarm of heterogeneous robotic agents. The method is based on the proposition that agents of different types experience different magnitude of potential while they are interacting with agents of different types. An analysis of the system with the proposed approach in Lyapunov sense is carried out for stability. Extensive simulation studies and numerical analysis suggest that the proposed method would always lead a population of heterogeneous agents closer to the sorted or segregated configuration.**

I. Introduction

Formation control of multiple autonomous vehicles has received attention of several researchers working in the area of mobile robotics because of its potential applications in a number of fields including cooperated search and rescue operation, surveillance, reconnaissance, and boundary protection. Advancement in communication and sensing technologies, and in computing resources have made it possible to coordinate the movement of several autonomous vehicles working cooperatively to achieve certain mission. One of the very first applications of formation control of multiple agents was behavioral simulation of flocks of birds, herd of animals and schools of fish for computer graphics by Reynolds [17]. He stated three simple behaviors that lead to flocking in birds and fish: collision avoidance, velocity matching, and flock centering (in decreasing order of precedence). The biggest merit of Reynolds' approach was that these behaviors were based on observations of local environment and interactions on a local scale that could be fully implemented in individual agents. These local interactions among agents resulted in global flocking, schooling, and herding behaviors which were totally scalable.

Drawing inspiration from Reynolds' approach, many researchers have focused on designing decentralized controller for achieving flocking behavior. The examples include behavior-based methods [2], leader-follower technique [5], [6], method based on formation constraint and virtual leaders/beacons [7], Lyapunov function based methods [13], [14], [15], [16]. The concept of artificial potential has been used in robotics by many researchers. For example, artificial potential has been used for path planning [24], manipulator control [23], robot navigation [12], and obstacle avoidance [11], and multi-robot formation control [13], [15].

Obtaining a desired shape and pattern of the formation can be critical for a mission relying on coordinated action by multiple mobile agents. For example, if a large number of robots need to be deployed to perform complicated tasks such as surveillance of large area, perimeter protection of vital installation, or surrounding site of a chemical or hazardous waste spill, the robots must be able to autonomously organize themselves in certain formation, pattern, or shape. In many situations, it may not be possible to integrate all the capabilities, sensing or actuation, required for different kinds of tasks in an individual robot. Accordingly, the robots may have heterogeneous abilities for sensing and actuation that will enable them to perform specific tasks. Heterogeneous robots must be able to self-organize themselves in a mission specific manner to carry out tasks assigned to them. The main contribution of this paper is synthesis and analysis of a controller that allows the robots to segregate or sort so that they form separate groups comprising of homogeneous robots.

II. Segregation and Sorting Behaviors in Biology

Sorting is a phenomenon which is seen in several biological systems. Examples include brood sorting by ants [4], segregation in amphibian larvae based on kinship [10], and aggregation /segregation behaviors in cockroaches [1] based on odors of strains. Sorting of cells based on their types and functionalities is one of the best examples of sorting in biological systems. Cell sorting is one of the basic phenomenon which leads to formation of patterns and organs in living organisms. Study of formation of patterns in living organisms is called morphogenesis. The mechanisms by which these patterns form can provide valuable insights for distributed problem solving strategies. Most of the strategies or models in literature that can explain formation of patterns rely on differential attraction/inhibition. For example, Swindale's model [20] accounts for formation of ocular dominance stripes in visual cortex based on local activation and lateral inhibition

This research was performed while the first author, Dr. Manish Kumar, held a National Research Council Associateship award through the Army Research Office.

He is currently with the Dept. of Mech., Indu., and Nuclear Engg. at the Univ. of Cincinnati, OH. manish.kumar@uc.edu

Dr. Devendra P. Garg is with the Dept. of Mech. Engg. and Matls. Sci. at Duke Univ., NC dpgarg@duke.edu

Dr. Vijay Kumar is with the Dept. of Mech. Engg. and App. Mech. at the Univ. of Pennsylvania, PA kumar@seas.upenn.edu

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

(LALI) mechanism [8] for like type of synapses, and local inhibation and lateral activation (LILA) [3], [20], the reverse, for the unlike type of synapses. Reaction-diffusion [22] is a model to mathematically represent the transport phenomenon in biological and natural systems. This model tries to explain the interaction of particles with the environment and their motion in space. In early 1990s, Graner and Glazier [9] proposed a lattice based modified version of large-Q Potts model with differential adhesivity to explain and simulate the sorting of a mixture of two types of biological cells. In fact, it has been long known [19] that it is the difference in intercellular adhesivity that leads to sorting in cells. The final state of cell configuration is achieved when the overall surface energy is globally minimized. Based on this principle, Steinberg [19] postulated that cells are sorted i.e., two types of cellular units A and B are segregated when:

$$W_{AB} < \frac{(W_{AA} + W_{BB})}{2} \tag{1}$$

where W_{AA} and W_{BB} represent the work of cohesion between particles or cells of same types (i.e., between types A & A, and B & B respectively), and W_{AB} represents the work of adhesion between cells of types A and B. The method for self-sorting in artificial mobile agents presented in this paper is motivated by this differential adhesivity phenomenon observed in biological systems that leads to sorting.

III. Problem Formulation

The group of mobile agents consists of N fully actuated agents, each of whose dynamics is given by the double integrator:

$$\begin{aligned} \dot{q}_i &= p_i \\ \dot{p}_i &= u_i(t) \qquad i = 1, 2, \ldots, N \end{aligned} \tag{2}$$

where q_i and p_i are m-dimensional position and velocity vectors respectively of agent i. The group of mobile agents, considered in this paper, consists of two different types of agents: type A and type B. The number of agents of type A is N_A and that of type B is N_B such that $N = N_A + N_B$. The objective of this paper is to synthesize a controller that can asymptotically flock and separate the robots of type A and type B into two different groups (referred to as *sorting* or *segregation*). Agents are said to flock (asymptotically) when all agents achieve the same velocity vector, distances between the agents are stabilized, and no collisions occur. Let us try to precisely define the term *Segregation*. A group of agents of types A and B are said to be *segregated* if there exists a hyperplane that separates the two different types of agents. Alternatively:

$$\exists \mathbf{e} : \langle (q_j - q_i), \mathbf{e} \rangle \geq 0 \qquad \forall i \in A \quad \& \quad j \in B \tag{3}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ means inner product between vectors **a** and **b**. This implies that every agent of type A is on one side of the hyperplane, and that of type B is on the other side of the hyperplane. For the sake of simplicity, we relax this definition and alternatively define *Segregation* to be a configuration of agents where the average distance between the agents of like types (type A or type B) is less than the average distance of agents between the unlike types (between agents of type A and type B). Alternatively,

$$r^{AA}_{avg} < r^{AB}_{avg}, \quad r^{BB}_{avg} < r^{AB}_{avg} \tag{4}$$

where r^{XY}_{avg} is the average distance between agents of types X and Y.

IV. Control Law Formulation

This section presents the control law which causes a population of heterogeneous agents to asymptotically flock as well as segregate. For a system of N mobile agents with N_A agents of type A and N_B agents of type B, following feedback control law is considered:

$$u_i = -\sum_{j \in N_i} \nabla_{q_i} V_{ij}(\|q_j - q_i\|) - a \sum_{j \in N_i} (p_i - p_j) \tag{5}$$

where u_i is the control input to the agent i, N_i is the set of agents in the neighborhood of agent i, $V_{ij}(\|q_j - q_i\|)$ is the artificial potential of interaction between agents i and j, $\|q_j - q_i\|$ is the norm of vector $(q_j - q_i)$ representing the euclidean distance between agent i and j, and ∇_{q_i} is the gradient with respect to coordinates of agent i i.e., q_i. First term in equation (5) represents the gradient of potential function, and the second term represents damping and causes the agents to match their velocities with each other. The artificial potential is a non-negative function of relative distances between a pair of neighbors given by $V_{ij}(q_i, q_j) : R^{2m} \rightarrow R_{\geq 0}$. Artificial potential function, V_{ij} due to interaction between two agents i and j can be expressed [13] as:

$$V_{ij} = \begin{cases} a\left(ln(q_{ij}) + \frac{d_0}{q_{ij}}\right) & \text{if} \quad 0 \leq q_{ij} \leq d_1 \\ a\left(ln(d_1) + \frac{d_0}{d_1}\right) & \text{if} \quad q_{ij} > d_1 \end{cases} \tag{6}$$

where, a is a scalar control gain, and $q_{ij} = \|q_j - q_i\|$. The parameters d_0 and d_1 respectively represent the inter-agent distance below which (i.e. when $q_{ij} < d_0$) the interaction force is repulsive (negative) and above which (i.e. when $q_{ij} > d_1$) the interaction force is zero. Figure 1 shows the potential function plotted against the inter-agent distance. As indicated in the figure, the potential becomes minimum when the inter-agent distance is d_0. The interaction between agents happen with the help of sensing or communication devices. The parameter d_1, then, can be regarded as the sensing or communication range. Without loss of generality, in this paper the parameter d_1 is considered infinity so that each agent can interact with the rest of the agents. Results presented in this paper remain unaffected for finite d_1 if connectedness of the underlying graph of the system is assumed. Equation (6), under such a condition, can be simply written as:

$$V_{ij} = a\left(ln(q_{ij}) + \frac{d_0}{q_{ij}}\right) \tag{7}$$

The basis for controller synthesis in this paper is the parameter d_0. Since there are two types of mobile agents

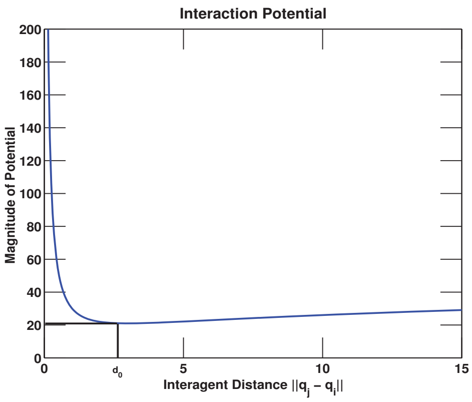


Fig. 1. Interaction Potential versus Inter-Agent Distance

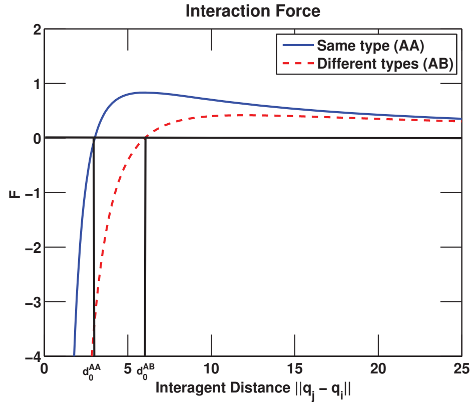


Fig. 2. Interaction Force between Agents

involved in the system, there are three different kinds of artificial potentials involved: a) Potentials arising due to interaction between types A and A, b) Potentials arising due to interaction between types B and B, and c) Potentials arising due to interaction between types A and B. In this paper, potentials arising due to interaction between the same types are considered to be same, i.e., $V_{ij}^{AA} = V_{ij}^{BB}$. The term V_{ij}^{AA} is given by:

$$V_{ij}^{AA} = a\left(ln(q_{ij}) + \frac{d_0^{AA}}{q_{ij}}\right) \tag{8}$$

and the term V_{ij}^{AB} is given by:

$$V_{ij}^{AB} = a\left(ln(q_{ij}) + \frac{d_0^{AB}}{q_{ij}}\right) \tag{9}$$

The control law for segregation can be achieved when:

$$d_0^{AA} = d_0^{BB} < d_0^{AB} \tag{10}$$

Figure 2 shows the plot of force of interaction due to similar types and due to dissimilar types of robots versus interagent distance when the condition for segregation controller (10) is met. In this case, it can be seen that the interaction force between agents of same types is greater than that of force between agents of different types at any given distance. Hence, this method of segregation, based on differential potential, is analogous to Steinberg's [19] explanation of cell sorting based on differential adhesiveness (see Equation (1)).

V. Controller Analysis

In this section, we carry out an analysis of convergence and stability properties of the system of multiple agents obeying dynamics given by Equation (2) under control law given by Equation (5). In order to carry out stability analysis of the collective motion of agents, the following positive definite function can be chosen as the Lyapunov function:

$$\phi(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{p} \tag{11}$$

where $\mathbf{q} \in R^{mN}$ is stacked position vector of all agents, $\mathbf{p} \in R^{mN}$ is stacked velocity vector of all agents, and $V(\mathbf{q}) : R^{mN} \rightarrow R_{\geq 0}$ is the total potential energy of the system.The total potential of the system consists of three parts: 1) Potential due to interaction of agents of type A, 2) Potential due interactions amongst agents of types A and B, and 3) Potential due to interaction of agents of type B. This can be written as:

$$\begin{aligned} V(\mathbf{q}) &= \mathbf{V_{AA}}(\mathbf{q}) + \mathbf{V_{BB}}(\mathbf{q}) + \mathbf{V_{AB}}(\mathbf{q}) \\ &= \frac{1}{2}\sum_{i\in A}\sum_{j\in A, j\neq i} V_{ij}(\|q_j - q_i\|) + \sum_{i\in A}\sum_{j\in B} V_{ij}(\|q_j - q_i\|) \\ &+ \frac{1}{2}\sum_{i\in B}\sum_{j\in B, j\neq i} V_{ij}(\|q_j - q_i\|) \end{aligned} \tag{12}$$

The collective dynamics of the system can be given by:

$$\dot{\mathbf{q}} = \mathbf{p} \tag{13}$$

$$\dot{\mathbf{p}} = -\nabla V(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{p} \tag{14}$$

where $\hat{L}(\mathbf{q}) \in R^{mN\times mN}$ is m-dimensional graph Laplacian (see reference [15]). Among other important properties of graph Laplacian matrix $\hat{L}(\mathbf{q})$, it is a positive semi-definite matrix.

Lemma 5.1: Consider a system of N mobile agents. Each of the agents follows dynamics given by Equation (2), and with feedback control law given by Equation (5). For any initial condition belonging to the level set of $\phi(\mathbf{q}, \mathbf{p})$ given by $\Omega_C = \{(\mathbf{q}, \mathbf{p}) : \phi(\mathbf{q}, \mathbf{p}) \leq C)\}$ with $C > 0$, and when the underlying graph of the system is connected and cohesive, then the system asymptotically converges to an invariant set $\Omega_I \subset \Omega_C$ such that the points in Ω_I have a velocity that is bounded and velocity of all agents match.

Differentiating $\phi(\mathbf{q}, \mathbf{p})$ with respect to time and using Equation (14) one gets:

$$\begin{aligned} \dot{\phi}(\mathbf{q}, \mathbf{p}) &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\dot{\mathbf{p}} \\ &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T(-\nabla V(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{p}) \\ &= -\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p} \leq 0 \end{aligned} \tag{15}$$

since $\hat{L}(\mathbf{q})$ is a positive semi-definite matrix. From Lasalle's Invariance Principle, all solutions of the system starting in Ω_C will converge to the largest invariant set

$\Omega_I = \Big\{(\mathbf{q},\mathbf{p}) \in \Omega_C : \dot{\phi}(\mathbf{q},\mathbf{p}) = 0)\Big\}$, and this happens when the velocity of all agents match. For a detailed proof of this lemma, please see references [13], [15]. Also, equilibrium condition is achieved when $\dot{\mathbf{p}} = 0$. Since there are no external forces acting on the agents, the velocities of all agents become zero (the center of mass of the system does not move), i.e. $\mathbf{p} = \mathbf{0}$. Hence, equilibrium is achieved when the total potential of the system is at extremum. This leads to the following proposition.

Proposition 5.2: When control actions lead to the global minimization of potential, then the system is segregated if the condition for segregation controller (10) is applied.

Since each term of $V(\mathbf{q})$ is a non-negative term, the global minimum is reached when each individual term of the expression (12) is minimum (assuming each term is independent of the other). This happens when:

$$\begin{aligned} r_{ij} &= \|q_i - q_j\| = d_0^{AA} \qquad \forall i \in A, j \in A \\ r_{ij} &= \|q_i - q_j\| = d_0^{BB} \qquad \forall i \in B, j \in B \\ r_{ij} &= \|q_i - q_j\| = d_0^{AB} \qquad \forall i \in A, j \in B \end{aligned} \tag{16}$$

Hence if condition (10) for controller is applied, we will have the segregation given by condition (4).

However, the system of agents with a global minimum configuration where all individual potentials are minimum is not achievable. Problems if this nature, referred to as graph embeddings [18], have been extensively studied in topological graph theory. More likely, the system will reach a local minimum condition which is given by:

$$\nabla V(\mathbf{q}) = 0 \tag{17}$$

The partial derivative given by equation (17) for an agent $i \in A$ is given by the equation:

$$\begin{aligned} \nabla_{q_i} V(\mathbf{q}) = &\sum_{j \in A, j \neq i} a \left[\frac{1}{\|q_i - q_j\|} - \frac{d_0^{AA}}{\|q_i - q_j\|^2} \right] \frac{(q_j - q_i)}{\|q_i - q_j\|} \\ &+ \sum_{j \in B} a \left[\frac{1}{\|q_i - q_j\|} - \frac{d_0^{AB}}{\|q_i - q_j\|^2} \right] \frac{(q_j - q_i)}{\|q_i - q_j\|} = 0 \end{aligned} \tag{18}$$

Equation (18) is also a force balance equation for an agent $i \in A$ and can be re-written as:

$$\sum_{j \in A, j \neq i} F_{ij}^{AA}(q_j - q_i) + \sum_{j \in B} F_{ij}^{AB}(q_j - q_i) = 0 \tag{19}$$

where

$$\begin{aligned} F_{ij}^{AA} &= \left[\frac{1}{\|q_i - q_j\|^2} - \frac{d_0^{AA}}{\|q_i - q_j\|^3} \right] \\ F_{ij}^{AB} &= \left[\frac{1}{\|q_i - q_j\|^2} - \frac{d_0^{AB}}{\|q_i - q_j\|^3} \right] \end{aligned} \tag{20}$$

If we sum up equation (19) for all $i \in A$, and noting that $F_{ij}^{AA} = F_{ji}^{AA}$, then we will obtain the following equation:

$$\sum_{i \in A} \sum_{j \in B} F_{ij}^{AB}(q_j - q_i) = 0 \tag{21}$$

Equation (21) leads to the following proposition:

Proposition 5.3: If we consider one dimensional case, i.e., $q, p \in R$, then the system of heterogeneous swarming agents following dynamics (2) and control law (5) flock together such that the average distance between the agents of different types is greater than or equal to the parameter d_0^{AB}, i.e., $r_{avg}^{AB} \geq d_0^{AB}$.

This proposition can be proved using equation (21) in the following manner. Let us write:

$$f_{ij}^{AB} = F_{ij}^{AB}(q_j - q_i) \tag{22}$$

Let us assume that out of $n_A n_B$ possible terms of f_{ij}^{AB}, m are the terms for which $q_{ij} = r_{ij} \geq d_0^{AB}$ and n ($m + n = n_A n_B$) are the terms for which $q_{ij} = r_{ij} < d_0^{AB}$. Hence, equation (21) can be written as:

$$\sum_{k=1}^{m} f_k^{AB} + \sum_{k=1}^{n} f_k^{AB} = 0 \tag{23}$$

Let us assume that $d_0^{AB} + x_c$ is the mean distance for the terms for which $r_{ij} \geq d_0^{AB}$, and $d_0^{AB} - x_{c'}$ is the mean distance for the terms for which $r_{ij} < d_0^{AB}$. Hence, equation (23) can be written as:

$$m\left(f^k|_{d_0^{AB} + x_c}\right) + n\left(f^k|_{d_0^{AB} - x_{c'}}\right) = 0 \tag{24}$$

Subsituting from equations (20) and (22):

$$\frac{m}{(d_0^{AB} + x_c)^2} x_c - \frac{n}{(d_0^{AB} - x_{c'})^2} x_{c'} = 0 \tag{25}$$

and since $(d_0^{AB} + x_c) \geq (d_0^{AB} - x_{c'})$, we have from equation (25):

$$m x_c \geq n x_{c'} \tag{26}$$

The average distance between agents of type A and type B is given by:

$$\begin{aligned} r_{avg}^{AB} &= \frac{1}{m+n} \left[m(d_0^{AB} + x_c) + n(d_0^{AB} - x_{c'}) \right] \\ &= d_0^{AB} + \frac{1}{m+n}(m x_c - n x_{c'}) \geq d_0^{AB} \end{aligned} \tag{27}$$

d_0^{AB} is a design parameter that can be chosen to be arbitrarily large value, and since r_{avg}^{AB} is always greater than d_0^{AB}, r_{avg}^{AB} can be made arbitrarily high.

Let us now examine the two-dimensional case when $q, p \in R^2$. Since F_{ij}^{AB} is a scalar quantity, in two dimensional case, equation (21) can be equivalently written into the following two scalar equations:

$$\sum_{i \in A} \sum_{j \in B} F_{ij}^{AB} x_{ij} = 0 \tag{28}$$

$$\sum_{i \in A} \sum_{j \in B} F_{ij}^{AB} y_{ij} = 0 \tag{29}$$

where $x_{ij} = (x_j - x_i)$ and $y_{ij} = (y_j - y_i)$, x_i and y_i are coordinates along X and Y axis for agent i. Hence $q_{ij} =$

$\sqrt{(x_{ij}^2 + y_{ij}^2)}$. Let us write equations (28) and (29) as the following:

$$\sum_{k:F_k \geq 0} F_k x_k + \sum_{k:F_k < 0} F_k x_k = 0 \qquad (30)$$

$$\sum_{k:F_k \geq 0} F_k y_k + \sum_{k:F_k < 0} F_k y_k = 0 \qquad (31)$$

If we assume:

$$\sum_{k:F_k \geq 0} F_k x_k = c_1 \qquad (32)$$

then

$$\sum_{k:F_k < 0} F_k x_k = -c_1 \qquad (33)$$

and similarly,

$$\sum_{k:F_k \geq 0} F_k y_k = c_2 \qquad (34)$$

$$\sum_{k:F_k < 0} F_k y_k = -c_2 \qquad (35)$$

Then from Cauchy-Schwarz inequality relation, equations (32) - (35), and noting that $q_k = \sqrt{x_k^2 + y_k^2)}$:

$$\|\mathbf{F}\|_{F_k \geq 0} \|\mathbf{Q^1}\| \geq \sqrt{c_1^2 + c_2^2} = c \qquad (36)$$

$$\|\mathbf{F}\|_{F_k < 0} \|\mathbf{Q^2}\| \geq \sqrt{c_1^2 + c_2^2} = c \qquad (37)$$

where $\mathbf{F} = \{F_1, F_2 ...\}$, $\mathbf{Q} = \{q_1, q_2 ...\}$, and $c \geq 0$.

Again from Cauchy-Shwarz inequality relations and equations (36) and (37), it is evident that:

$$0 \leq \sum_{F_k \geq 0} F_k q_k \leq c \qquad (38)$$

$$-c \leq \sum_{F_k < 0} F_k q_k \leq 0 \qquad (39)$$

From inequalities (38) and (39), we can write:

$$-c \leq \sum_{F_k \geq 0} F_k q_k + \sum_{F_k < 0} F_k q_k \leq c \qquad (40)$$

Using similar analysis used to prove Lemma (5.3), we can show that:

$$m x_c - n x_{c'} \geq -c \qquad (41)$$

where m is the number of terms for which $f_k \geq 0$, and n is the number of terms for which $f_k < 0$. This leads to the following:

$$r_{avg}^{AB} \geq d_0^{AB} - \frac{c}{m+n} \qquad (42)$$

The value c in the above inequality is bounded because the quantities c_1 and c_2 in equations (32) and (34) are bounded. Since c is bounded, we can always choose d_0^{AB} to be arbitrarily large making r_{avg}^{AB} to be arbitrarily large.

VI. Simulation Results and Discussions

Extensive simulations were carried out to verify the results obtained in the previous sections. In the simulations, following parameters were assumed :
$d_0^{AA} = d_0^{BB} = 3$, and $d_0^{AB} = 6$

Figure 3 shows the configuration of a population of 20 agents (10 each of type A and B) in a 2D space at different times during the simulation. The agents started off at a random configuration, and control law given by equation 5 based on differential potential was applied to the agents. The final configuration at time T=750 sec shows that the agents of types A and B form two separate groups.

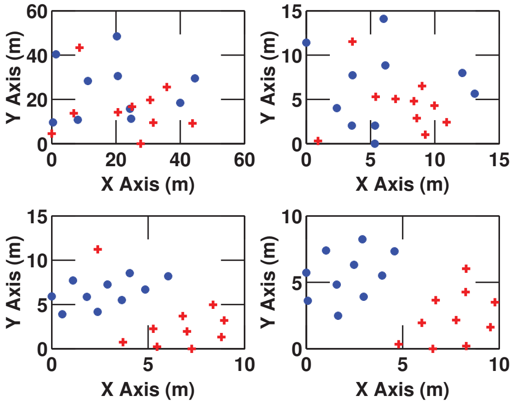


Fig. 3. Configurations of Agents at Times T=0 (top left), T=250 sec (top right), T=500 (bottom left), and T=750 sec (bottom right)

Figure (4) shows the plot of average distances between agents of types A and A (r_{avg}^{AA}), B and B (r_{avg}^{BB}), and A and B (r_{avg}^{AB}) versus time for the above simulation. At the final configuration, the average distances r_{avg}^{AA}, r_{avg}^{BB}, and r_{avg}^{AB} were found out to be 3.21, 3.22, and 6.97 respectively, which clearly shows that the population was segregated based on the condition given by (4).

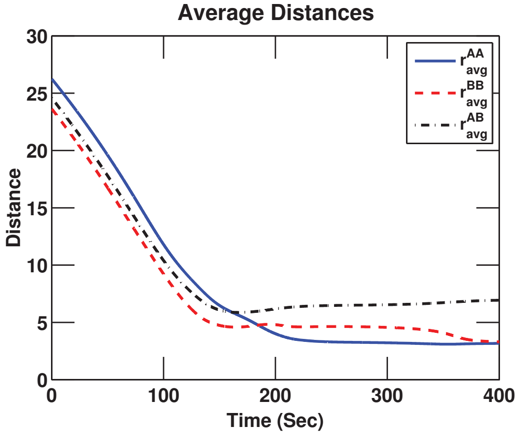


Fig. 4. Average Distances Between Agents of Types A and A (r_{avg}^{AA}), B and B (r_{avg}^{BB}), and A and B (r_{avg}^{AB})

The results given above were for just one simulation run. In order to verify that the method presented in this paper leads to segregation in general in a population of heterogeneous agents, an extensive simulation study was carried out in which more than 100 runs were performed. Figure 5 shows the average distance between agents at steady-state. In each of the simulation runs, the population of agents consisting of types A and B was initialized in a random configuration obtained via uniform distribution of agents in 2D space, and number of agents of type A and B were each chosen randomly between values 5 and 15. Each of the runs was carried out for 500 seconds of simulation time. The average distances between agents shown in the figure are calculated at the steady (final) state. It can be easily seen that the average distance between agents of type A (r_{avg}^{AA}) and average distance agents of type B (r_{avg}^{BB}) is less than the average distance between agents of type A and B (r_{avg}^{AB}) for each of the simulation runs. Also, it is evident from the figure that r_{avg}^{AB} is always greater than the parameter d_0^{AB} supporting our result from proposition 5.3. Moreover, in each of the simulation runs, the agents were completely segregated and satisfied the separating hyperplane condition of segregation given by equation (3).

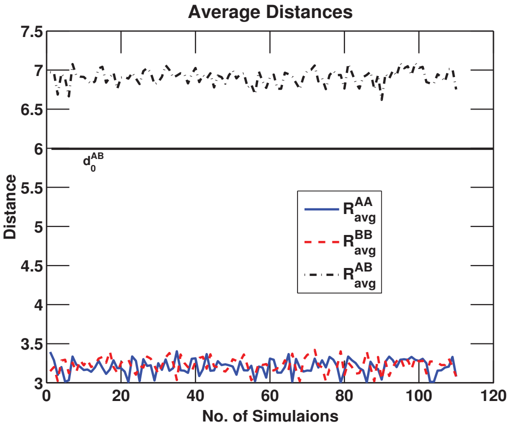


Fig. 5. Average Distance between Agents

The method presented in this paper is equally applicable to more than two types of agents and in higher dimensional space.

VII. Conclusions

The paper presents a decentralized method to achieve self organized bahavior of sorting or segregation in a population of heterogeneous agents. The method is based on the concept of differential artificial potential. The paper presents the stability analysis of a population of agents in Lyapunov framework, and lays down an analytical foundation for synthesis of controllers for self-sorting in artificial potential function framework. Specifically, condition for the synthesis of controllers for sorting is analyzed in one and two dimensional space. Extensive simulation studies verify the results obtained in this paper, and shows the effectiveness of the proposed method in achieving the sorting behavior in a swarm of heterogeneous agents.

References

[1] J. Ame, C. Rivault, and J. Deneubourg. Cockroach aggregation based on strain odour recognition. *Animal Behaviour*, 68:793–801, 2004.

[2] T. Balch and R. Arkin. Behavior-based formation control for multirobot systems. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[3] E. Bonabeau. From classical models of morphogenesis to agent-based models of pattern formation. *Artificial Life*, 3:191–211, 1997.

[4] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chreitien. The dynamics of collective sorting: robot-like ants and ant-like robots. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 353–363. MIT Press, Cambridge, MA, 1991.

[5] J. P. Desai, J. P. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2864–2869, 1998.

[6] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[7] M. Egerstedt and X. Hu. Formation constrained multi-agent control. In *IEEE International Conference on Robotics and Automation*, pages 3961–3967, 2001.

[8] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12:30–39, 1972.

[9] F. Graner and J. A. Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical Review Letters*, 69(13):2013–2016, Sept. 1992.

[10] M. A. Halverson, D. K. Skelly, and A. Caccone. Kin distribution of amphibian larvae in the wild. *Molecular Ecology*, 15:1139–1145, 2006.

[11] J. O. Kim and P. K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349, 1992.

[12] D. E. Koditschek. Exact robot navigation by means of potential functions: some topological considerations. In *IEEE Conference on Robotics and Automation*, volume 4, pages 1–6, 1987.

[13] N. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *IEEE International Conference on Decision and Control*, pages 2968–2973, 2001.

[14] P. Ogren, M. Egerstedt, and X. Hu. A control lyapunov function approach to multi-agent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.

[15] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, Mar. 2006.

[16] R. Olfati-Saber and R. M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *IEEE Conference on Decision and Control*, volume 3, pages 2965–2971, 2002.

[17] C. Reynolds. Flocks, birds, and schools:a distributed behavioral model. *Computer Graphics*, 21:25–34, 1987.

[18] I. J. Schoenberg. On certain metric spaces arising from Euclidean spaces by change of metric and their embedding in Hilbert space. *Annals of Mathematics*, 38(4):787–793, Oct. 1937.

[19] M. S. Steinberg. Reconstruction of tissues by dissociated cells. *Science*, 141:401–411, 1963.

[20] N. V. Swindale. A model for the formation of ocular dominance stripes. *Philosophical Transactions of the Royal Society of London B*, 208:243–264, 1980.

[21] H. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *IEEE International Conference on Decision and Control*, pages 2010–2015, 2003.

[22] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237(641):37–72, 1952.

[23] R. Volpe and P. Khosla. Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1423–1436, 1990.

[24] C. W. Warren. Global path planning using artificial potential fields. In *IEEE Conference on Robotics and Automation*, pages 316–321, 1989.

A.2.2 Role of Stochasticity in Self-Organization of Robotic Swarms

The following paper (on the next 6 pages) was co-authored by Manish Kumar, Dejan Lj. Milutinović, and Devendra P. Garg, and appeared in the Proceedings of the 2008 American Control Conference on pages 123–128, published and presented in June 2008.

2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Role of Stochasticity in Self-Organization of Robotic Swarms

Manish Kumar, Dejan Milutinović and Devendra P. Garg

Abstract— **This paper investigates the effectiveness of designed random behavior in self-organization of swarm of robotic agents. Inspired by the self-organization observed in biological cells and the role played by random forces in providing robustness in cell self-organization, we investigate the possibility of designing a decentralized controller for a swarm of agents in which the stochastic process is included. This paper considers flocking as a self-organizing behavior example to validate our findings. The controller is designed in the framework of Lyapunov function, and it is based on the artificial potential due to interactions among agents. Our analysis shows that the flocking behavior of the swarm is improved and is more robust when the stochastic process is included in the agent controller.**

I. Introduction

A swarm of robotic agents performing coherent activities while controlled in a decentralized manner is an example of a self-organized system [12]. There are several examples in nature where individual units carry out operations based on local interaction and local information without a complete knowledge of other units' operation. Yet the overall emerging behavior of the system appears to be highly organized, coherent, and efficient in achievement of its objectives. The best examples of such systems are populations of biological cells that possess the ability to self-organize into specific formations, form different types of organs and, ultimately, develop into a living organism. Most importantly, their ability to self-organize is extraordinarily robust.

The cell behavior is guided by biochemical signals and the structure of the environment. Intracellular biochemical signaling networks are involved in the detection of the environment and they drive the cell behavior, their function and motility. The role of signaling networks for cells is similar to the one that the robot hardware plays for the algorithms that guide the robot communication and sensory systems, as well as the robot behavior [10]. Because of this, the study of self-organized cellular systems is more relevant for the design of robotic swarms as compared to the study of biological multi-agent systems, such as flocks of birds or schools of fish, in which the presence of natural intelligence among biological entities cannot be ruled out. In this respect, the design of agent swarms inspired by a self-organized cellular system is a very promising direction of research.

This paper investigates the role random forces can play in attaining robust behaviors in swarm systems. The paper draws inspiration from the role of randomness in several search algorithms and in several biological systems. Randomness has been widely used as a component in several stochastic search algorithms. For example, evolutionary search algorithms [5] have heavy stochastic components (e.g. choice of chromosomes for reproduction, crossover, and mutation are driven by probabilities based on their fitness function). Reinforcement learning techniques [18] have two components: exploitation and exploration. Exploration is a random walk in the search space that makes the algorithm investigate new regions. Apart from randomness being traditionally used in search and optimization algorithms, random components have shown to play critical role in modeling many biological systems. Bateson [3] calls the mind a stochastic system and cognitive learning process a stochastic process. Contemporary cognitive scientists consider mental processes as stochastic processes such as evolutionary algorithms where hypotheses or ideas are proposed, tested, and either accepted or rejected by a population. Random or trial-and-error learning techniques provide ways to create new varieties of solutions for problems. Random behavior is ubiquitous in biological systems. Chaotic behavior of a hooked fish, random behavior among preys for predator avoidance, and zig-zagging of a chased rabbit through a meadow are all examples of existence and heavy use of random behaviors among animals. Lorenz [9], in his intuitive chapter entitled "Oscillation and Fluctuation as Cognitive Functions", has described the importance of a random behavior in organisms' motion for search, as well as for escaping dangers. We use this kind of random behavior as a component in the control laws for swarm robotic systems.

In this paper, we consider flocking which is the most investigated example of swarm self-organization. *Flocking* is a self-organized behavior in which agents, initially distributed over the operating space, group together and organize into a specific formation. An example of this kind of flocking is the formation of germinal centers inside the lymph node [16]. Flocking has been extensively studied in multi-agent literature. Reynolds [15] has been able to reproduce, in his computer models, behaviors representing flocking in birds and schooling in fish using simple rules based on local interactions among agents. Drawing inspiration from Reynolds' approach, many researchers have focused on designing a decentralized controller for achieving flocking behavior [2], [4], [13], [19]. A control system based on the methods presented in these references should be able to yield a single flock of agents based on local information. However, if

This research was performed while the first author, Dr. Manish Kumar, held a National Research Council Associateship award through the Army Research Office. He is currently with the Dept. of Mech., Indu., and Nuclear Engg. at the Univ. of Cincinnati, OH. manish.kumar@uc.edu

Dejan Milutinović is with the Laboratory for Computational Immunology at Duke University, NC dejan.m@duke.edu

Dr. Devendra P. Garg is with the Dept. of Mech. Engg. and Matls. Sci. at Duke Univ., NC dpgarg@duke.edu

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

the local information can only be collected over a finite range, it can lead to the formation of more than one flock. This paper studies scenarios in which local interaction leads to the fragmentation of groups and investigates how the introduction of random processes in an agent controller helps to eliminate or alleviate the problem of fragmentation.

In this paper, we consider flocking in the Lyapunov function framework. The Lyapunov function for the controller design of a multi-agent system is based on the artificial potential function of agent interactions [8], [14]. Being inspired by the role played by the random fluctuation in driving self-organizing processes in cells, we propose to include a random process in the decentralized controller for an agent swarm. Consequently, we analyze the swarm self-organization using the stochastic Lyapunov function [7]. We show that our proposed decentralized controller provides the robust flocking of the swarm of agents.

This paper is organized as follows. We introduce a model of swarming agents in section II. In section III, we describe a deterministic decentralized design of controllers providing flocking in a population of agents. Dealing with the problem of local minima, we suggest including a stochastic process into the controller in section IV. In section V, we derive emergent behavior related to the center of the mass of the flock. Section VI presents numerical simulations illustrating the performance of the controller, and finally section VII presents the conclusions derived from the present work.

II. Modeling of swarm system and Lyapunov function definition

The group of mobile agents consists of N fully actuated agents, each of whose dynamics is given by the double integrator:

$$\begin{aligned} \dot{q}_i &= p_i \qquad (1)\\ \dot{p}_i &= u_i(t) \qquad i = 1, 2, \ldots, N \end{aligned}$$

where q_i, p_i and u_i are m-dimensional position, velocity and control vectors of agent i, respectively. This double integrator representing particle dynamics [14], [8], [20] is a popular and realistic model to represent the motion of agents in a multi-agent system. It facilitates the implementation of decentralized control algorithms, and provides a mechanism to include limitations due to sensing and communication as compared to other models, such as continuum model [6], [11], which represents the collective motion of agents in the form of particle density functions. The model given in equation (1) can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \qquad (2)$$

where vector $\mathbf{x}(t) = [\mathbf{q}(t)\ \mathbf{p}(t)]^T$ and

$$\mathbf{q}(t) = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix},\ \mathbf{p}(t) = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix},\ \mathbf{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \qquad (3)$$

and matrices $\mathbf{A}$ and $\mathbf{B}$ are

$$\mathbf{A}(\mathbf{x}, t) = [\mathbf{I}_{N\times N}\ \mathbf{0}_{N\times N}]^T,\ \mathbf{B} = [\mathbf{0}_{N\times N}\ \mathbf{I}_{N\times N}]^T. \qquad (4)$$

In order to carry out a stability analysis of the collective motion of agents, a Lyapunov function can be chosen as the total energy

$$\phi(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{p} \qquad (5)$$

The Lyapunov function is composed of the total artificial potential energy $V(\mathbf{q})$ and the kinetic energy, the second term of the sum given in equation (5). We define the potential energy as a non-negative function

$$V(\mathbf{q}) = \frac{1}{2}\sum_{i=1}^{N} V_i = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i} V_{ij}(\|q_i - q_j\|) \qquad (6)$$

$$V_{ij}(\|q_i - q_j\|) \geq 0 \qquad (7)$$

where V_i is the total artificial potential associated with an agent i. This energy depends on the set N_i comprised of the agents inside the communication range of the agent i. The artificial potential function V_{ij} depends upon the Eucledian distance $\|q_i - q_j\|$ between the agent i and j, and it is given by:

$$V_{ij} = \begin{cases} a(ln(\|q_i - q_j\|) + \frac{d_0}{\|q_i - q_j\|}), & 0 \leq \|q_i - q_j\| \leq d_1 \\ a(ln(d_1) + \frac{d_0}{d_1}), & \|q_i - q_j\| > d_1 \end{cases} \qquad (8)$$

where, a is a scalar parameter. The parameters d_0 and d_1 respectively represent the inter-agent distance below which the interaction force is repulsive (negative) and above which the interaction force is zero. Figure 1 shows the interaction potential plotted against the inter-agent distance. It can be easily seen that the potential becomes minimal when the inter-agent distance is d_0. The interaction among agents happens with the help of sensing or communication devices. The parameter d_1, then, can be regarded as a sensing or a communication range.

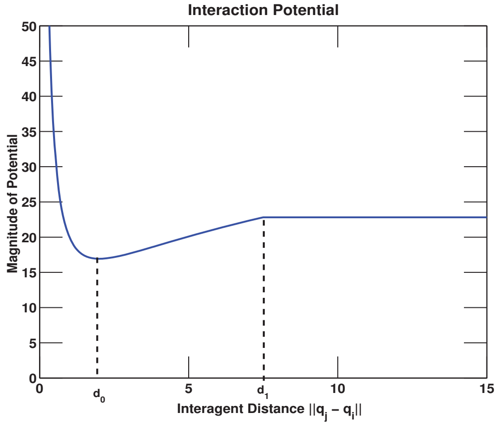


Fig. 1. Interaction potential versus inter-agent distance

III. Controller Design for Flocking

Flocking is a form of self-organized behavior of agent swarms in which agents meet or come together. The collective dynamics of the system can be analyzed using a

Lyapunov function. We can differentiate $\phi(\mathbf{q},\mathbf{p})$ with respect to time, and using expression (5) and model (1), one gets:

$$\begin{aligned}\dot{\phi}(\mathbf{q},\mathbf{p}) &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\dot{\mathbf{p}} \\ &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\mathbf{u}(t)\end{aligned}$$

Using arguments based on the Lyapunov function stability analysis, the state configuration $\mathbf{p}$, $\mathbf{q}$ will be stable if

$$\dot{\phi}(\mathbf{q},\mathbf{p}) = \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\mathbf{u}(t) \leq 0 \tag{9}$$

The control law ensuring this type of stability is described by the following lemma of which a detailed proof is provided in references [14] and [8].

Lemma 3.1: Consider a system of N mobile agents. Each of the agents follows dynamics given by model (1) and with the feedback control law given by

$$u_i = -\nabla_{q_i} V_i + f_i^v \tag{10}$$

where ∇_{q_i} represents the gradient with respect to position q_i of agent i, $c > 0$ is a scalar gain and f_i^v is given by:

$$f_i^v = c\sum_{j\in N_i}(p_j - p_i) \tag{11}$$

For any initial condition belonging to the level set of $\phi(\mathbf{q},\mathbf{p})$ given by $\Omega_C = \{(\mathbf{q},\mathbf{p}) : \phi(\mathbf{q},\mathbf{p}) \leq C)\}$ with $C > 0$, and when the underlying graph of the system is connected and cohesive, the system asymptotically converges to an invariant set $\Omega_I \subset \Omega_C$ such that the points in Ω_I have a velocity that is bounded and the velocities of all agents match.

To illustrate this lemma, it is worth mentioning that the control law (10) results from the fact that it can be equivalently described in the vector form as

$$\mathbf{u}(t) = -\nabla V(\mathbf{q}) - \hat{\mathbf{L}}(\mathbf{q})\mathbf{p} \tag{12}$$

where $\hat{L}(\mathbf{q}) \in R^{mN\times mN}$ is m-dimensional graph Laplacian (see reference [14]), which is a positive semi-definite matrix. Obviously, this control satisfies condition (9). Also, from Lasalle's Invariance Principle, all solutions of the system starting in Ω_C will converge to the largest invariant set $\Omega_I = \left\{(\mathbf{q},\mathbf{p}) \in \Omega_C : \dot{\phi}(\mathbf{q},\mathbf{p}) = 0)\right\}$, and this occurs when the velocities of all agents match.

IV. Controller design based on the stochastic Lyapunov function

Using the deterministic Lyapunov function controller design, agents always move in such a way that the covered distances and their directions do not increase the Lyapunov function. It is ultimately expected that the agents reach a stable formation (flock) in which the Lyapunov function attains its extremum value. If the agents that we are dealing with have a limited communication range, it is possible that robots reach stable formations with more than one cluster. In this configuration, agents from one cluster are out of the communication range of any agent from the other clusters. The graph Laplacian $\hat{L}(\mathbf{q})$ is no longer positive semi-definite in this case, and a local minimum of the Lyapunov function is reached.

Here, we introduce a controller which includes the random process term, providing means to an escape from local minima. Using the notation of section II, the model of a robot population can be written in the matrix form as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{\Sigma}\xi(t) \tag{13}$$

where

$$\mathbf{\Sigma} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & diag(\sigma_1,\sigma_2,\ldots\sigma_N) \end{array}\right]_{2N\times 2N} \tag{14}$$

and the controller effect is composed of the part which is assumed to be computed based on the local agent information $\mathbf{B}\mathbf{u}(t)$ and the stochastic part $\mathbf{\Sigma}\xi(t)$.

Using the Lyapunov function as before and using the Îto formula, we can find the time derivative of the Lyapunov function

$$\dot{\phi} = \sum_i p_i\frac{\partial\phi}{\partial q_i} + u_i(t)\frac{\partial\phi}{\partial p_i} + \frac{1}{2}\sum_i \sigma_i^2\frac{\partial^2\phi}{\partial p_i^2} + \sum_i \sigma_i\frac{\partial\phi}{\partial p_i}\xi_i(t) \tag{15}$$

that results in

$$\begin{aligned}\dot{\phi} = \; & \textstyle\sum_i p_i\frac{\partial V_i(q_1,q_2,\ldots q_N)}{\partial q_i} + \sum_i u_i(t)p_i + \\ & \textstyle\frac{1}{2}\sum_i\sigma_i^2 + \sum_i\sigma_i p_i\xi_i(t)\end{aligned} \tag{16}$$

The latter expression means that in this case the Lyapunov function ϕ of the robotic population is a stochastic process.

Similar to the deterministic case, if we would like to design a controller which aligns the robot velocities, i.e., provides flocking, we can define u_i as

$$u_i = -\frac{\partial V_i(q_1,q_2,\ldots q_N)}{\partial q_i} - [\hat{L}(\mathbf{q})\mathbf{p}]_i \tag{17}$$

Under this condition

$$\dot{\phi} = -\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p} + \frac{1}{2}\sum_i\sigma_i^2 + \sum_i\sigma_i p_i\xi_i(t) \tag{18}$$

which means that the total Lyapunov function value is a stochastic process. The stochasticity provides escape from the local minima. We assume, without losing generality, that $\sigma_i = \sigma$. The intensity of this stochastic process is governed by the parameter σ which needs to be determined by taking appropriate considerations of factors explained below.

There is a stochastic steady-state (see reference [7], page 50, Theorem 6) for the value of ϕ in which the following condition is satisfied:

$$E\left\{\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p}\right\} = \frac{1}{2}\sum_i\sigma_i^2 = \frac{N\sigma^2}{2} \tag{19}$$

There are two limits for σ that should be avoided. One is $\sigma < \sigma_L$ that results in the deterministic controller. In this case the swarm does not flock robustly. The second limit is $\sigma > \sigma_H$, when σ is large and leads to a large expected value of $\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p}$. This means that the robot velocities are poorly aligned. For reasons of the robust flocking, σ should be in the range of σ_L and σ_H. Out of this range, the flocking does not happen robustly due to the small ($\sigma < \sigma_L$) or large

($\sigma > \sigma_H$) intensity of the random process, respectively. The value of σ has to be tuned taking into account the constraints of the actuator that drives the robots and it is a part of the controller design. Introduced controller does not guarantee the attainment of global minimum of the Lyapunov function, but it can guarantee that the formation of the system is, most of the time, around this minimum.

V. Emergent Behavior of a Stochastic Control for Flocking

The cluster is a group of robots in which each robot is in the communication range of at least one robot. Motion of each agent i obeys the following stochastic differential equation

$$\begin{aligned} \dot{q}_i &= p_i && (20)\\ \dot{p}_i &= u_i(t) + \sigma_i \xi_i(t) \ . && (21) \end{aligned}$$

This expression includes two parts: the first part based on the local information and the second part based on the stochastic process of intensity σ_i. We assume again that ξ is the Gaussian white noise of the unit intensity. For reasons of simplicity, but without losing generality, we will consider that $\sigma_i = \sigma$.

Since each robot has a unique label i, the cluster C_j can be defined as the set of labels i of the robots being in the cluster j. Assuming that the cluster exists within the time interval $\tau = [t_1 \ t_2]$, the motion of the C_j cluster's center of the mass is described by

$$\begin{aligned} \dot{q}_j^{CM} &= p_j^{CM} && (22)\\ \dot{p}_j^{CM} &= \frac{1}{|C_j|}\sum_{i \in C_j} \sigma_i \xi_i(t) = \frac{\sigma}{\sqrt{|C_j|}}\xi(t) && (23) \end{aligned}$$

where the number of the robots in the cluster is denoted by $|C_j|$. The terms dependent on u_i are cancelled due to the symmetry in the interaction between any two robots in the cluster. The sum of white Gaussian noises is also the white Gaussian noise and the last term in (23) includes this properly scaled unit intensity Gaussian noise.

From (23), we can conclude that the center of the cluster mass moves randomly through the operating space. The smaller the cluster, the faster it "explores" the operating space before it meets other robots to form a larger cluster. We quote the word "explore", because it is a behavior that emerges from the inclusion of the random process in the robot controller.

Obviously, while exploring, the cluster may decompose into two smaller clusters. The probability of cluster decomposition can be made very small by appropriately choosing the intensity σ to be small in comparison to the intensity of u_i. Regardless of this, the cluster decomposition does not influence our further analysis, because we analyze the steady-state property of the robot formation independently of how it is reached. In the limit when $|C_j|$ is large, the center of the cluster mass will move with $q_j^{CM} \approx 0$.

VI. Simulation Results

Before we present the simulation results, let us explain *social* and *hierarchical social* entropy metrics we use to quantify the degree of the self-organization of agent populations. Social entropy, inspired by Shannon's information entropy [17], has been used in multi-agent systems [1] as a metric for diversity in behavior or properties of agents, including diversity in their spatial locations. This metric captures an important feature of diversity, which is the number of differentiable groups, i.e., clusters, in a system and the size of these clusters.

We define the *cluster* as a group of agents in which the distance between the agent and its closest neighbor from the same cluster is smaller than some threshold value h, which is the communication range distance. If an agent does not have the neighbor within a distance smaller that h, then this agent is alone in the cluster.

If a group of n agents, represented by the set R, can be divided into m clusters $C_1, C_2 \ldots C_m$, such that there is no agent which is simultaneously in two clusters, then the social entropy H of the system R is given by

$$H(R) = -\sum_{i=1}^{m} \frac{|C_i|}{|R|} \log \frac{|C_i|}{|R|} \qquad (24)$$

with $|.|$ representing the number of agents in the cluster and $\sum_i |C_i| = |R|$, where $|R| = n$ is the number of robots in the group.

The social entropy H obviously depends on the threshold h. In order to develop a measure that can accurately represent the diversity in a population of agents regardless of the scale or the value h, *hierarchical social entropy* has been used in literature [1], which is given by:

$$E(R) = \int_{h=0}^{\infty} H(R,h)dh \qquad (25)$$

where $H(R,h)$ is the social entropy given by equation (24).

To illustrate flocking in the absence and the presence of the random force, we simulate a group of 20 agents during a 500 sec. time span. In our simulations, the agents have a limited communication range, which is modeled with the artificial potential function parameter $d_1 = 20m$ (see equation (8)). The other parameter of this function is $d_0 = 6m$, and the intensity of the random process $\sigma = 0.25$. In the simulation scenario, the agents are allowed to move only in the 100×100 rectangular area (see Fig. 2). To define this area inside the simulation, we introduce an additional artificial potential function resulting in short-range repulsive forces making agents move away from the area border.

Figure 2 shows the agent configurations at different points of time of the simulation when random force in the controller is not applied. The agents start from an initial randomly chosen position. By the terminal time of the simulation $T = 500$ sec, the agents form three separate clusters, which are the consequence of the limited communication range. In the next figure, Fig. 3, we illustrate the simulation in which the agents start from the same configuration as in the previous example, but now the random force inside the controller is applied. We note that by the terminal simulation time the agents form a single cluster.

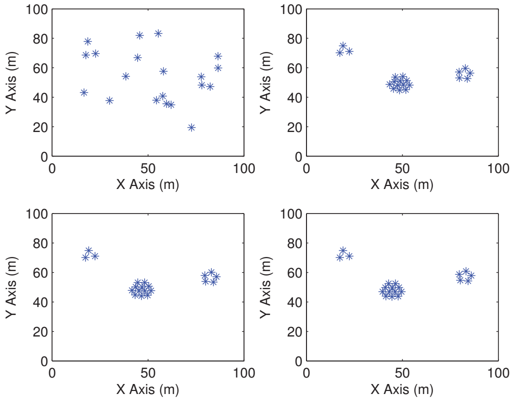


Fig. 2. Configurations of agents at times T=0 (top left), T=167 sec (top right), T=333 sec (bottom left) and T=500 sec (bottom right) when no random motion is applied

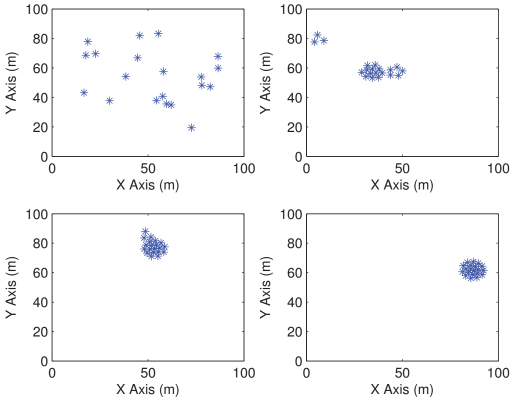


Fig. 3. Configurations of agents at times T=0 (top left), T=167 sec (top right), T=333 sec (bottom left) and T=500 sec (bottom right) when random motion is applied

The hierarchical social entropy, given by equation (25), for these two examples is plotted against the simulation time in Fig. 4. From this figure, it is evident that the application of random force results in lower entropy values, meaning that agents aggregate closer to one another forming bigger clusters.

In order to verify that the introduction of random motion does lead to better flocking behavior of agents, we carried out two sets of 100 simulation runs. Each run was carried out under the same conditions as in the two presented examples. In one simulation set, the random force term of the controller is applied and in the other, it is not. For each simulation run we computed the social entropy at the terminal time $T = 500$ sec. The distribution of the computed social entropy values, for each set of simulations, is presented in the form of the cumulative distribution function shown in Fig. 5. It can be seen that the increase in the number of simulation occurrences for the case when random motion is not applied is slower than that of the case when random motion is applied. This is because of the fact that most of the simulations with random motion have an entropy which is smaller as compared to simulations with no random motion. A smaller entropy indicates a smaller (spatial) diversity, i.e., better flocking behavior.

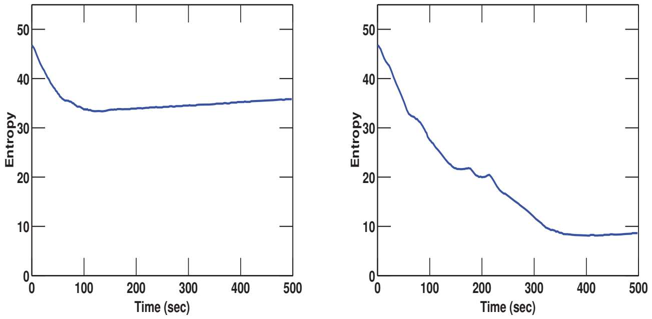


Fig. 4. Hierarchical social entropy of the population of agents for the case when no random motion is applied (left), and the case when random motion is applied (right)

The ultimate evidence that the application of the random force term leads to a better flocking behavior is the histogram of number of simulations with specific number of clusters in the final configuration for the two cases, as shown in Fig. 6. As can be seen from the figure, a lesser number of clusters are formed for that case when random motion is applied as compared to the case when random motion is not applied. The average number of clusters for the case when no random motion is applied is 3.02, while the average number of clusters when random motion is applied is 1.51. This shows a marked improvement in flocking behavior and the formation of one giant cluster as compared to the case when no random motion is applied. It may be noted that the separate clusters formed in the second case move randomly in a confined space. This random motion within a confined space results into the probability of clusters finding each other approaching unity when the time approaches infinity.

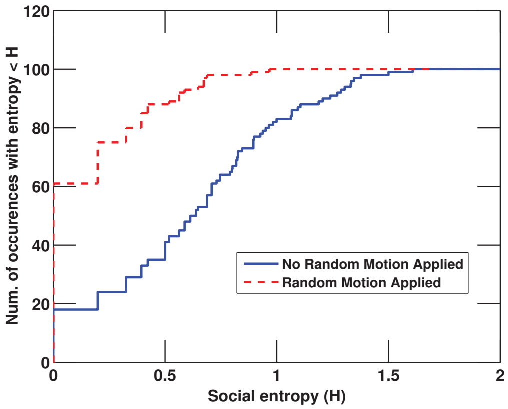


Fig. 5. Simple social entropy at final configuration of agents for 100 simulation runs

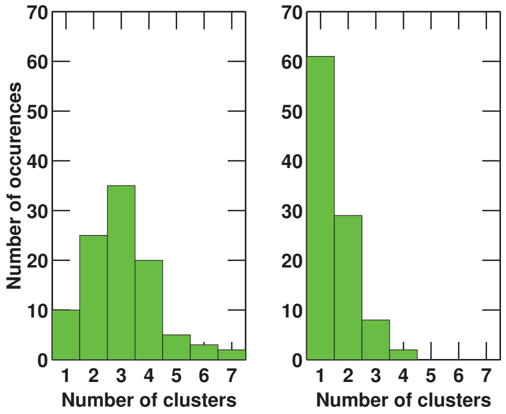


Fig. 6. Number of clusters of agents at final configuration for 100 simulation runs when no random motion is applied (left) and when random motion is applied (right)

VII. Conclusion

Inspired by the role of random forces in robustness of self-organization of cells and other biological systems, we investigated the possibility of designing a decentralized controller for a swarm of agents in which a stochastic process is included. We considered the flocking behavior of a swarm and described the previously considered deterministic Lyapunov function controller design based on the artificial potential of interactions among agents. The deterministic design may not lead to desired swarm behaviors because of the existence of agent configurations in which the total potential of swarm interactions has local minima.

The design we considered is an extension of the deterministic one. We used the same Lyapunov function. However, due to the introduced stochastic process, the Lyapunov function is also a stochastic process. Consequently, this controller provides a means of escaping from the local minima of the total potential of the swarm interactions.

The intensity of the included stochastic process is of great significance. If its intensity is small, then the swarm behavior is the same as if controlled by the deterministic controller. If the intensity is high, then the flocking behavior cannot be established because the swarm can even escape from the region around the global minimum. Only if the intensity of the stochastic process lies within some intermediate range, does the swarm escape the local minima and randomly explores configurations, which can ultimately lead the swarm towards configurations close to the global minimum of the total potential of interactions. The drawback of this design is that even in the global minimum configuration, the stochastic process forces the swarm to search for a better configuration. Tuning the intensity of the stochastic process of the controller is identical to consideration that appears in the design of the realistic model of cell behavior, or to the problem of providing physiological conditions for self-organization in biological cells.

The performance of the proposed controller was illustrated by the derivation of the emergent swarm behavior and an extensive simulation study.The performance of the controller with the random process term was compared to the performance of the deterministic controller. The simulation results suggest that the inclusion of the random process in the controller can certainly improve the performance of the robotic swarm in achieving the flocking behavior.

References

[1] T. Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3):209–237, 2000.

[2] T. Balch and R. Arkin. Behavior-based formation control for multirobot systems. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[3] G. Bateson. *Mind and Nature*. New York: E. P. Dutton, 1979.

[4] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[5] D. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley Longman, Inc., 1989.

[6] E. W. Justh and P. S. Krishnaprasad. Steering laws and continuum models for planar formations. In *IEEE International Conference on Decision and Control*, pages 3609–3614, 2003.

[7] H. J. Kushner. *Stochastic Stability and Control*. Academic Press, New York, 1967.

[8] N. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *IEEE International Conference on Decision and Control*, pages 2968–2973, 2001.

[9] K. Lorenz. *Behind the Mirror: A Search for a Natural History of Human Knowledge*. New York: Harcourt Brace Jovanovich, 1973.

[10] D. Milutinović and P. Lima. *Cells and Robots : Modeling and Control of Large-Size Agent Populations*. Springer, 2007.

[11] D. Milutinović, P. Lima, and M. Athans. Biologically inspired stochastic hybrid control of multi-robot systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.

[12] G. Nicolis and I. Prigogine. *Self-organization in non-equilibrium systems*. Wiley, 1977.

[13] P. Ogren, M. Egerstedt, and X. Hu. A control Lyapunov function approach to multi-agent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.

[14] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, Mar. 2006.

[15] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, 21(4):25–34, 1987.

[16] T. A. Schwickert, R. L. Lindquist, G. Shakhar, G. Livshits, D. Skokos, M. H. Kosco-Vilbois, M. L. Dustin, and M. C. Nussenzweig. In vivo imaging of germinal centres reaveals a dynamic open structure. *Nature*, 446:83–87, 2007.

[17] C. E. Shannon and W. Weaver. *A Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL, 1949.

[18] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction, MIT Press*. MIT Press, 1998.

[19] H. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *IEEE International Conference on Decision and Control*, pages 2010–2015, 2003.

[20] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, May 2007.

A.2.3 Mobile Multi-Robot Control in Target Search and Retrieval

The following paper (on the next 7 pages) was co-authored by Guoxian Zhang and Devendra P. Garg, and appeared in the Proceedings of the 2008 ASME Dynamic Systems and Control Conference on pages 677–683, published and presented in October 2008.

Proceedings of DSCC2008
2008 ASME Dynamic Systems and Control Conference
October 20-22, 2008, Ann Arbor, Michigan, USA

DSCC2008-2196

MOBILE MULTI-ROBOT CONTROL IN TARGET SEARCH AND RETRIEVAL

Guoxian Zhang
Mechanical Engineering & Materials Science Dept.
Duke University
Durham, North Carolina 27708
Email: guoxian.zhang@duke.edu

Devendra P. Garg[*]
Mechanical Engineering & Materials Science Dept.
Duke University
Durham, North Carolina, 27708
Email: dpgarg@duke.edu

ABSTRACT

In this paper, the design of a controller is proposed for a multi-robot target search and retrieval system. Inspired by research in insect foraging and swarm robotics, we developed a transition mechanism for the multi-robot system. Environmental information and task performance obtained by the robot system are used to adjust individual robot's parameters and guide environment exploration. The proposed control system is applicable in the solution of multi-target problem also where several robots may be needed to cooperate together to retrieve a large target. Simulations show that the task performance improves significantly with the proposed method by sharing information in parameter learning and environment exploration.

INTRODUCTION

The interest in developing robots that can cooperate together to perform tasks is increasing, since some tasks may be difficult, inefficient, or impossible for a single robot to perform [1]. The cooperative intelligence of animals can be seen in nature especially in social insects, such as ants and bees [2]. These insects cooperate to search for prey and retrieve them to their nest with only indirect communication and local sensing ability among members and the environment. Inspired by these insects, a group of simple robots may use cooperation to perform target location and retrieval [3–6]. The research on multi-robot foraging has a variety of possible applications such as demining, toxic odor source search, and treasure hunting [7], where robots' cooperation may increase the system's performance.

The foraging behavior of ants was studied by Holldobler and Wilson [8] with two characteristics: ants randomly explore the environment to search for prey; and the prey is dragged by individual or collective retrieval straight to the nest according to its weight. Three kinds of learning foraging (local performance-based reinforcement, global performance-based reinforcement, and local shaped reinforcement) were discussed by Balch in [9]. In a macro view, differential equations can be used to model the flow of robot state changes in the system [10]. An analytical and spatial swarm robots foraging model using virtual pheromone based partial differential equations was discussed by Hammann and Worn in [11]. Labella and Dorigo [5] attempted to improve the performance of robot foraging by optimizing each robot's probability of changing state from rest to search under its previous performance in target searching and retrieving. Experience of other members was ignored.

Liu [3] utilized environmental cues, internal cues, and social cues to change each robot's threshold for search and rest based on all robots' performance in foraging, including interference between robots and obstacles. This method can improve the system's net energy income compared to systems without adaption. Net energy income is the energy value of the total retrieved targets minus total system energy consumption. However, researchers Ren and Williams [12] indicated that communication is not necessary in multiagent learning for robot foraging.

Different from decentralized control, Ogras, Dagci and Ozguner [13] presented a hierarchical control architecture in cooperative control for multi-robot target search. This method needs a supervisor in the system to decompose the goal into tasks and delegate them to robots. Besides foraging, multi-robot system is also studied in coverage problem. A review of methods on robot coverage was given by Choset in [14]. Hazon and Kaminka [15] showed that redundant coverage could guarantee better performance in coverage time than non-redundant coverage. Wagner, Lindenbaum, and Bruckstein [16] proposed an ant-robots system using smell traces that gradually vanish with time. This method

[*] Address all correspondence to this author.

could complete the traversal even when some of the members die or the graph changes. In some applications, such as treasure hunting and odor source search, methods in foraging, retrieval and coverage may be combined to improve the system performance.

In some tasks, such as demining, different kinds of targets may be deployed in an area and the target density decreases during the process. In this paper, we develop a method for a group of robots performing multiple target search and retrieval tasks. Targets with different values and weights are initially deployed in an unknown environment. The robot system is required to search in the unknown environment to find targets and try to retrieve them to the nest within a certain time period. Only local communication among robots and local sensing of the environment are assumed to be available during the entire task. The effect of communication among robots is studied and used to adjust robots' parameters and search process.

PROBLEM FORMULATION

This paper addresses the problem of generating a control strategy for a robot group involved in target search and retrieval tasks. The robots are initially deployed in a specific area called the nest, denoted by $\mathcal{N}$, in an area $\mathcal{A}$. All of the robots are assumed to have the same properties. For example, they have limited sensor field of view denoted by $\mathcal{S}$, and load carrying ability w_0. The workspace is populated with n fixed obstacles $\{\mathcal{B}_1,\ldots,\mathcal{B}_n\}\subset\mathcal{A}$, and m targets $\{\mathcal{T}_1,\ldots,\mathcal{T}_m\}\subset\mathcal{A}$. Besides location, each target $\mathcal{T}_i$ has two other properties, value v_i and weight w_i.

In order to gather the targets into the nest, robots need to first search in $\mathcal{A}$ to locate the targets, and when a target is found, the robot may choose, according to the target's weight, to pull it into the nest individually or ask for recruitment. According to the energy and time constraint, there may not be enough time for the robots to locate and retrieve all the targets in $\mathcal{A}$ to $\mathcal{N}$. The problem then changes to achieving an optimal group performance in the given time period T. The group performance can be measured by the net energy gain of the group. When a target $\mathcal{T}_i$ is retrieved to $\mathcal{N}$, the energy gain $G_{\mathcal{T}_i}$ of the retrieval can be considered as the difference between the target value v_i and the total energy consumption c_i to retrieve this target. In this problem, the energy consumption is assumed to be proportional to the robots' traveling distance

$$c_i = k_r k_0 d \tag{1}$$

where d is the retrieval distance, k_0 is the coefficient relating the distance and energy consumption in retrieval, and k_r is the number of robots retrieving the target. Then the net energy gain can be represented as

$$G_{\mathcal{T}_i} = v_i - k_r k_0 d \tag{2}$$

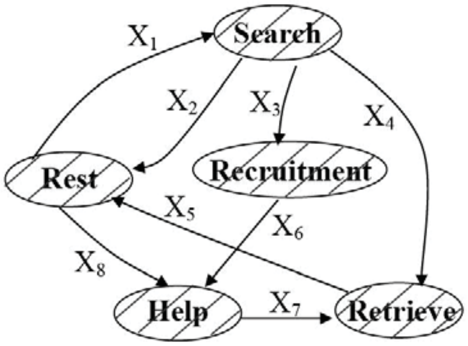


Figure 1. TRANSITION DIAGRAM OF THE ROBOTS

Besides energy consumption in the retrieval task, the robot system's energy consumption may also occur in robots that are searching, recruiting, and even resting. Considering the summation of the consumption above as c_0, and if I denotes the set of indices of targets which have been retrieved successfully, the total net energy income of the robot system which can be considered as the performance index is

$$G = \Sigma_{i\in I} G_{\mathcal{T}_i} - c_0 \tag{3}$$

The final object of this problem is to improve the value of G in a given time period T while searching the proper area and retrieving proper targets.

PROPOSED APPROACH

Inspired by the observation of ants foraging and considering the time constraint in our problem, we assume that all the robots have the same properties, such as moving speed, load carrying ability, and sensor range. The energy is consumed by robots in the system with different rates according to their states, for instance searching and resting. Since in this problem targets are deployed at the beginning, each robot can utilize the environmental information obtained both by itself and other members to adjust its parameters and improve the search efficiency. With finite state machine strategy the transition diagram of the foraging task is illustrated in Fig. 1. The control of the foraging task is decomposed into several states. The states of the robot include:

Rest: The robot stays in the nest.

Search: The robot randomly explores the environment in order to find targets.

Recruitment: The robot moves back towards the nest in order to ask for help from other members to retrieve the target it has found.

Help: The robot moves to the located target with some other robots in order to retrieve it together.

Retrieve: The robot carries the target towards the nest.

The arrows in Fig. 1 show the possible transitions between robot states. The possibility of a robot to change to another state

Table 1. VARIABLES IN ROBOTS STATES TRANSITION AND CONTROL

Variable	Description
N	total number of robots in the system
N_r	number of robots in rest state
N_s	number of robots in search state
N_{rt}	number of robots in retrieve state
N_{rc}	number of robots in recruitment state
N_h	number of robots in help state
E_r	energy consumption of robot in rest per second
E_s	energy consumption of robot in search per second
E_{rt}	energy consumption of robot in retrieve per second
E_{rc}	energy consumption of robot in recruitment per second
E_h	energy consumption of robot in help per second
X_i	number of robots flow between two states $(i = 1,\ldots,8)$

depends on the performance of the robot system and the environment information it has obtained. The variables in this control graph are shown in Tab. 1

At time t, the change of robots in each state can be represented by X_is at time t, for example

$$dN_r(t) = -X_1(t) + X_2(t) + X_5(t) - X_8(t) \tag{4}$$

If the indices of targets retrieved successfully up to time t is denoted by the set $I(t)$, the energy gain at time t can be represented as:

$$\begin{aligned} dG(t) = \quad & d\Sigma_{i\in I(t)} v_i - dN_r(t)E_r - dN_s(t)E_s \\ & -dN_h(t)E_h - dN_{rt}(t)E_{rt} - dN_{rc}(t)E_{rc} \end{aligned} \tag{5}$$

The second, third, and fourth terms are corresponding to the change of c_0 at time t in eq. (3). N_{rt} and N_{rc} are corresponding to k_r in eq. (2). E_{rt} and E_{rc} are corresponding to $k_0 d_0$ in eq. (2), where d_0 represents robot's traveling distance per second. The purpose of this exercise is to improve $G(T)$ in a given time period $[0,T]$. In this paper, we try to control X_is during the time period $[0,T]$ to obtain a large net energy income for the entire time period. The control strategy of these transition variables are as follows:

$X_1(t)$: Define a threshold $T_r(t)$ which decreases when a target is retrieved by a robot or other members and increases when a robot or other members come back to the nest with no target found. When the robot has been in rest state not less than $T_r(t)$, it changes to search state.

$X_2(t)$: Define a threshold $T_s(t)$ which increases when a target is retrieved by a robot or other members and decreases when a robot or other members come back to the nest with no target found. When the robot has been in search state not less than $T_s(t)$, it changes to a temporary homing state and when it arrives at the nest it changes to rest states. $T_s(t)$ is updated for a robot when it is about to leave the nest to search and is kept constant during the robot's entire foraging period.

$X_3(t)$: Define a threshold $T_{rc}(t)$ which averages the energy gain of the robot system over time. When a large target is first found, and the net energy gain to ask teammates to retrieve it is higher than $\rho T_{rc}(t)$, where ρ is a constant, the robot changes its state to recruitment.

$X_4(t)$: When a robot finds a target which is movable by it alone, it changes its state to retrieve.

$X_5(t)$: When a robot successfully retrieves a target to the nest, it changes its state to rest.

$X_6(t)$: When a robot in recruitment arrives at the nest and finds enough teammates, it changes to help state. Otherwise, if there are not sufficient teammates in the nest, it changes to a temporary wait state until the required number of teammates is met.

$X_7(t)$: When a robot in the help state arrives at the target, it changes to a temporary wait state, and then changes to retrieve state when all the members arrive.

$X_8(t)$: When a robot is in the rest state and is asked by a robot in recruitment state, it changes to a temporary wait state. When there are enough members it changes to help state.

We assume that the communication among robots can only happen when they are in the nest. By this assumption, the diversity of information held by each robot is small and may make the control strategy concise. In this case, the parameters, such as T_r, T_s and T_{rc}, for each robot can only be adjusted when it is in the nest. When the robot is foraging, all of these parameters are kept constant. Parameters for different robots may be different due to their different time to leave the nest.

The obtained environmental information can also be used to guide the robot's search process. First the searching field is equally divided into n regions as shown in Fig. 2. We assume that each robot has the ability to remember the environment it has explored. The search strategy can be represented as follows:

1. When each robot arrives at the nest back from foraging, it shares its exploration experience with other robots in the nest.

2. When a robot changes to search state, it chooses the region which has been explored by fewest number of robots based on the robot's learned information. Before leaving the nest, the robot informs all the robots in the nest the region it has chosen.

3. A random unknown area in the chosen region is generated as the primary goal, and the robot searches towards the goal area first. If no target is found, then it continues its random search in the field.

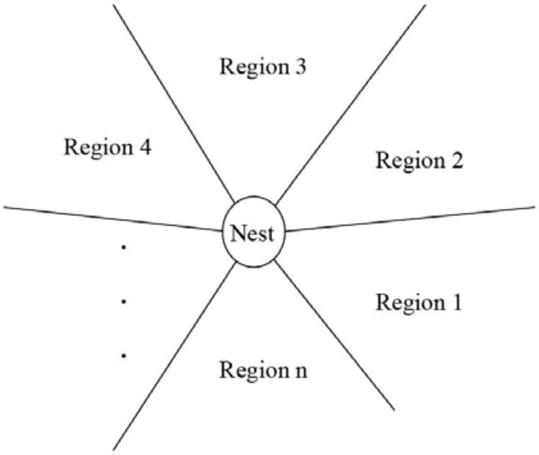


Figure 2. REGION DIVISION OF THE FIELD

Table 2. TYPICAL TARGET PROPERTIES

Type	1	2	3	4	5	6	7
Weight	1	2	2	2	4	4	4
Value	400	400	800	1200	800	1600	2400

5. When searching, the robot preferentially covers the area which has not been explored by other robots according to its obtained information. If all the possible moving directions are occupied by obstacles or other robots, the robot chooses to stay until a free place appears.

The performance of the control strategy mentioned above is simulated as described in the following section.

SIMULATION EXAMPLE

In this section, a simulation system is developed to test the method proposed above. The following assumptions are made in the simulation.

(1) The time is discretized into small finite steps.

(2) The space and targets are decomposed to be squares.

(3) Each robot is assumed to take the shape of a square with the ability to sense the eight squares adjacent to it and load carrying ability of one weight unit.

(4) The information of the members, such as their success in retrieval and area already explored, can be transferred among robots. The information is transferred if and only if both the sender and the receiver are in the nest.

(5) The robot can move up, down, left or right one square at each time step.

(6) Targets are assumed to occupy one square, with typical values and weights specified in Tab. 2.

Table 3. ENERGY CONSUMPTION PER STEP IN EACH STATE

E_r	E_s	E_{rt}	E_{rc}	E_h	E_{ho}	E_w
0.2	1	4	0.6	0.6	0.6	0.2

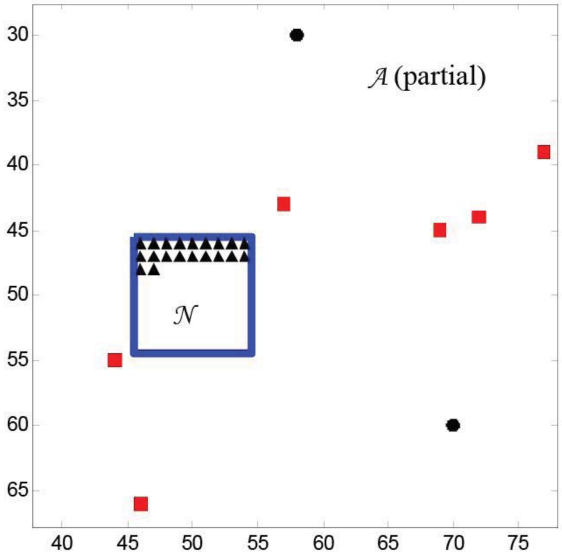


Figure 3. AN EXAMPLE OF SIMULATION FIELD REGION

(7) Energy consumption is assumed as shown in Tab. 3

The simulation field is shown in Fig. 3. The boundary of the nest is represented by wide lines; robots are shown as triangles; obstacles are shown as circles, and targets are shown as rectangles.

The control for each robot is discretized at each time step. When a robot reaches the nest, it may communicate with other robots in the nest and share the information about the field it has explored and its current task performance. Every robot at rest state computes its T_r based on the updated information, and if timeout, it changes to search state and computes its T_s and T_{rc} by the obtained information before leaving the nest. A control flow diagram is shown in Fig. 4.

An example of the simulation field at 200th time step is shown in Fig. 5. In this figure we can see that one target is retrieved by four robots and another target is retrieved by a single robot.

We create simulations under two conditions to examine the performance of our method.

1. Robot uses shared information to adjust its parameters and guide its search process.

2. Robot uses its own information to adjust its parameters and guide its search process.

All the robots are assumed to be located in the nest at the beginning. At the first step all of them change to search state. After

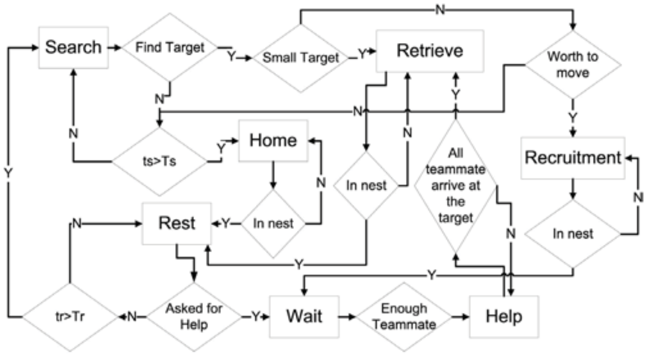


Figure 4. FLOW CHART FOR EACH ROBOT

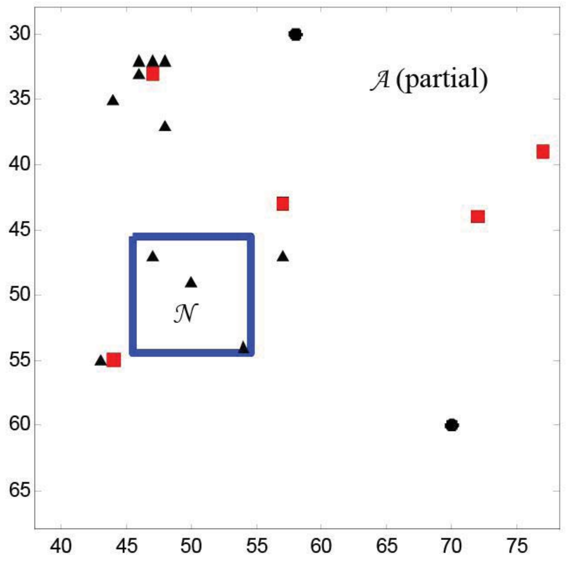


Figure 5. AN EXAMPLE OF SIMULATION FIELD REGION AT 200TH TIME STEP

the first robot comes back to the nest from foraging, it changes to a special ”queen” state. The robot at ”queen” state will stay in the nest for the entire remaining time period and share information with a robot which has just arrived at the nest or is about to leave the nest. This way, any robot returning to the nest may have access to the most updated and complete information. This allows information in the nest to be consolidated. Each simulation has a total of 1000 time steps, and all the results are the average of 50 runs. Net energy gains of the systems with 10, 15 and 20 robots under conditions 1 and 2 are shown in Tab. 4.

The net energy gain versus time for a total robot population of 15 is shown in Fig. 6; the number of robots in search state versus time is shown in Fig. 7; and the number of robots in the nest versus time is shown in Fig. 8.

Table 4. NET ENERGY GAIN UNDER DIFFERENT CONDITIONS

Condition	1	2
E_{10}	6493	568
E_{15}	5862	-1265
E_{20}	4914	-2549

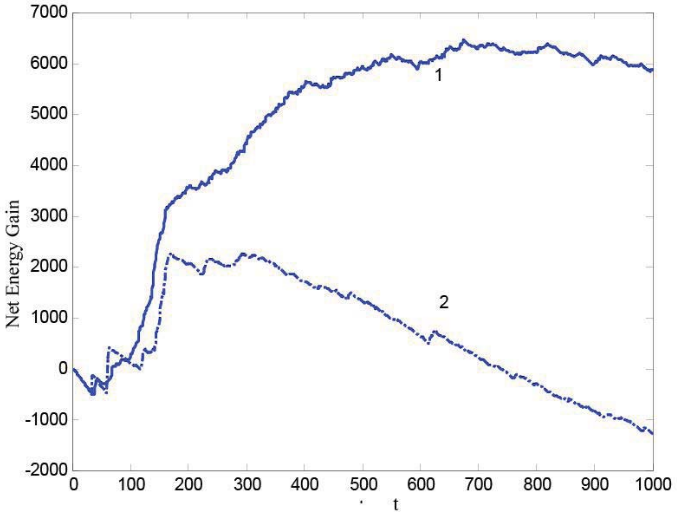


Figure 6. NET ENERGY GAIN IN MULTI-TARGETS FORAGING

From the results in Tab. 4 and Fig. 6, we can see that the use of members’ information on the environment is important to assess the system performance. Without sharing information the system’s net energy income decreases at about the 300th step. The possible reason is that without knowing other members’ exploration information, it is highly possible that the robot will continue to search into a place which has been explored by other robots. This would waste the system energy and time. For both conditions, the net energy gain decreases as the number of robots increases. With finite targets deployed in the field, the higher the number of robots, the higher is the system’s energy consumption. Moreover, the system with higher number of robots has a higher probability of interference among robots, targets, and obstacles. This will decrease the efficiency of foraging and retrieving. Fig. 7 and Fig. 8 show that under condition 1 fewer robots are in search state most of the time and more robots are in rest or wait state in the nest. These results are consistent with those shown in Tab. 4. From Tab. 4, since the net energy decreases as the number of robots increases, we can see that the system with 15 robots seems to have more robots than the task needs. By communicating and sharing the information, the system under condition 1 divides robots into different states with fewer robots foraging and more robots in the nest, which achieves a lower system energy consumption. This shows the system under condition 1 has a better adaption to the environment.

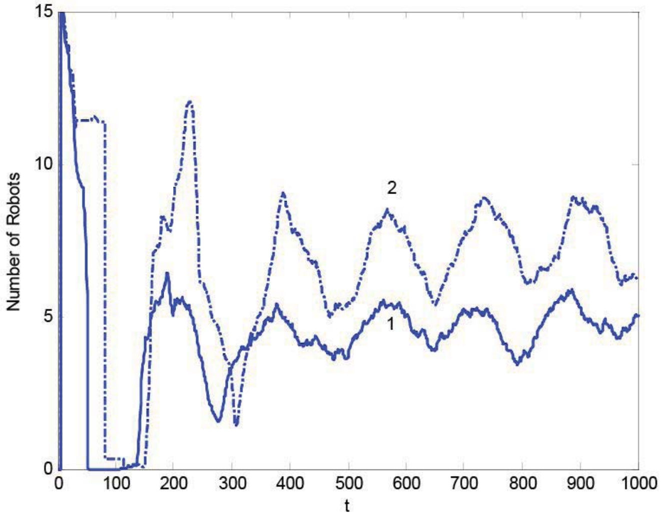


Figure 7. ROBOTS IN SEARCH

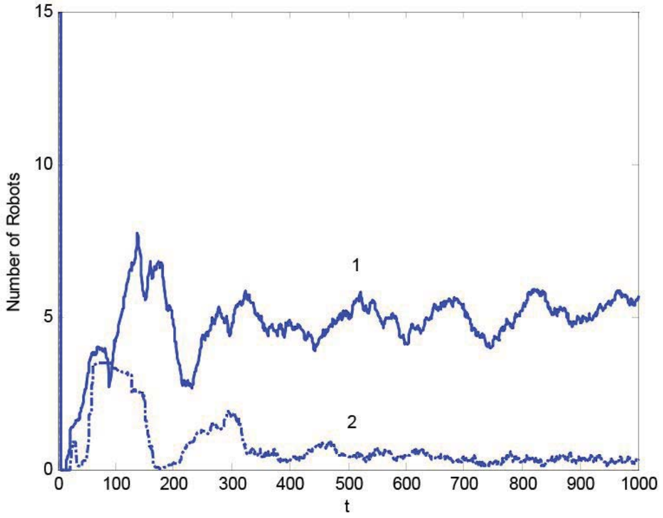


Figure 8. ROBOTS IN THE NEST

CONCLUSION

In this paper, we have presented a methodology for multi-robot system used in a target foraging and retrieval problem. Finite State Machine is used to control the robot behaviors. Information obtained by the individual robots is shared in the nest and is utilized to adjust each robot's parameters in state transitions. Shared information is also used to create a strategy which helps the robots to choose a search region and guide their movement during foraging. Simulation results show that communication of individual robot's experience on task and field searching could improve the system net energy gain up to 10 times in chosen conditions compared to the case without sharing information and keep at a high system net energy gain when the later one decreases significantly.

ACKNOWLEDGEMENT

The support provided by the U.S. Army Research Office under grant number W911NF-08-1-0106 is gratefully acknowledged.

REFERENCES

[1] Bekey, G. A., 2005. *Autonomous Robots*. MIT Press, Cambridge, Massachusetts, USA.

[2] Bonabeau, E., Dorigo, M., and Theraulaz, G., 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York.

[3] Sahin, E., ed., 2007. *Swarm Robotics*, Vol. 4433 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Germany, Chap. 2, pp. 14–26.

[4] Ulam, P., and Balch, T., 2004. "Using optimal foraging models to evaluate learned robotic foraging behavior". *International Society for Adaptive Behavior,* **12**(3), September, pp. 213–222.

[5] Labella, T. H., and Dorigo, M., 2006. "Division of labor in a group of robots inspired by ants' foraging behavior". *ACM Transactions on Autonomous and Adaptive Systems,* **1**(1), September, pp. 4–25.

[6] Lerman, K., Jones, C., Galstyan, A., and Mataric, M., 2006. "Analysis of dynamic task allocation in multi-robot systems". *The International Journal of Robotics Research,* **25**(3), pp. 225–241.

[7] Sahin, E., ed., 2005. *Swarm Robotics*, Vol. 3342 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Germany, Chap. 2, pp. 10–20.

[8] Holldobler, B., and Wilson, E., 1990. *The Ants*. Springer Verlag, Heidelberg, Germany.

[9] Sahin, E., ed., 2005. *Swarm Robotics*, Vol. 3342 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Germany, Chap. 3, pp. 21–30.

[10] Costa, F., ed., 2007. *Advances in Artificial Life*, Vol. 4648 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Germany, pp. 696–705.

[11] Sahin, E., ed., 2007. *Swarm Robotics*, Vol. 4433 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Germany, Chap. 4, pp. 43–55.

[12] Ren, Z., and Williams, A., 2003. "Lessons learned in single-agent and multiagent learning with robot foraging". In Systems, Man and Cybernetics, IEEE International Conference, Vol. **3**, pp. 2757–2762.

[13] Ogras, U., Dagci, O., and Ozguner, U., 2004. "Cooperative control of mobile robots for target search". In Mechatronics, IEEE International Conference on Mechatronics, pp. 123–128.

[14] Choset, H., 2001. "Coverage for robotics-a survey of recent

results". *Annals of Mathematics and Artificial Intelligence,* **31**, pp. 113–126.

[15] Hazon, N., and Kaminka, G., 2005. "Redundancy, efficiency and robustness in multi-robot coverage". In Robotics and Automation, IEEE International Conference, pp. 735–741.

[16] Wagner, I., Lindenbaum, M., and Bruckstein, A., 1999. "Distributed covering by ant-robots using evaporating traces". *IEEE Transactions on Robotics and Automation,* **15**(5), October, pp. 918–933.

A.2.4 Sensing and Estimation on a Modular Testbed for Swarm Robotics

The following paper (on the next 8 pages) was co-authored by Gregory K. Fricke, Devendra P. Garg, and Dejan Lj. Milutinović, and appeared in the Proceedings of the 2009 ASME Dynamic Systems and Control Conference, Volume 2, on pages 195–202, published and presented in October 2009.

Proceedings of the ASME 2009 Dynamic Systems and Control Conference
DSCC2009
October 12-14, 2009, Hollywood, California, USA

DSCC2009-2723

SENSING AND ESTIMATION ON A MODULAR TESTBED FOR SWARM ROBOTICS

Gregory K. Fricke, Devendra P. Garg[*]
Robotics and Manufacturing Automation Laboratory
Mechanical Engineering and Materials Science Department
Duke University
{gkf4, dpgarg}@duke.edu

Dejan Milutinović
Applied Mathematics and Statistics Department
University of California at Santa Cruz
dejan@soe.ucsc.edu

ABSTRACT

Collective robotics offers the promise of enhanced performance and robustness relative to that of individual robots, with decreased cost or time-to-completion for certain tasks. Having many degrees of freedom, the multi-robot control and estimation problems are challenging, specifically when the solutions require a great amount of communication among the robots. While numerical simulation is a critical tool in swarm robotics research, verification of obtained results under a physical realization of the swarm is far from routine. Therefore, we have developed and used a sensor-integrated testbed for the validation of cooperative-robotics algorithms, observation of swarm behavior, and measurement of system performance.

INTRODUCTION

Conducting experiments in swarm robotics is challenging, time-consuming, and expensive, leading many investigators in the field to extensively rely on computer simulations. While simulation is indispensable for algorithm development and performance estimation, the physical realization of a swarm system is important for full understanding and validation of swarm behavior. Also, physical experiments shed light on the nature of uncertainty sources in the sensors, actuators, and communication pathways specific to mobile robotic systems while illustrating the logistical difficulties involved in implementing swarm behavior.

Robot pose estimation is critical for feedback control of robot trajectory. Methods for pose estimation and robot localization have been extensively studied, utilizing methods such as dead-reckoning [1–6], on-board and off-board computer vision [7,8], environmental sensing [9,10], collective or distributed localization [11, 12], and simultaneous localization and mapping (SLAM) [13] using laser-scanning systems [14, 15], on-board cameras [16], RFID tags and readers [17,18], or ultrasonic rangefinders [10].

Pose estimation and tracking of individual robots are critical to the metrology required for complete evaluation of experimental results. In the case of multi-robot systems, the scalability of the metrology system is a critical concern. With multiple robots, reliance on advanced vision-based techniques quickly surpasses all but the most advanced computational systems, imposing the restriction that performance data must be evaluated off-line without real-time constraints.

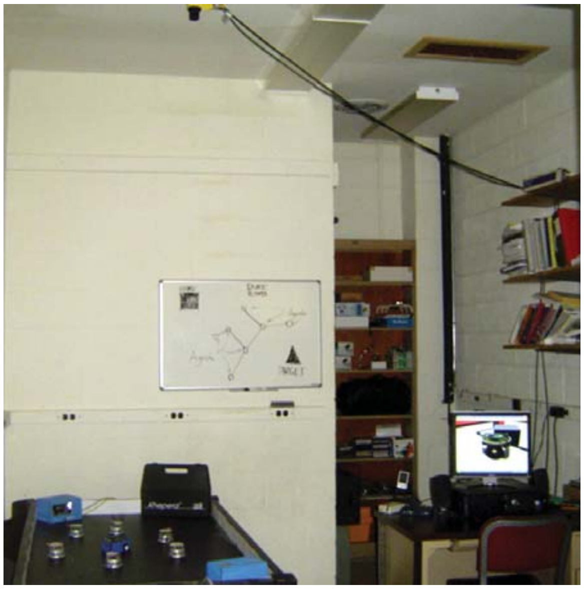


Figure 1. Khepera-II arena of the RAMA Lab.

[*] Address all correspondence to this author.

The modular testbed described in this paper has been developed at the Robotics and Manufacturing Automation (RAMA) Laboratory at Duke University. The testbed, shown in Fig. 1, utilizes multiple Khepera-II robots operating within a defined swarm arena equipped with two SICK LMS400 LADAR sensors and an overhead Cognex Insight 5400 industrial machine-vision system. Researchers at other institutions have recently described similar multi-robot testbeds, e.g., see [15, 19, 20]. Our focus has been to maximize utility, maintain scalability, and avoid overly specialized and cost-prohibitive equipment.

The paper is organized as follows. First, the swarm members, LADAR units, and vision sensor are described. Next, the estimation methods developed to utilize the data are discussed, and the communication methods employed are presented. The final sections are devoted to the results of simulated experiments using real raw data, including a discussion of plans for future development.

SWARM MEMBERS

The swarm members in use are Khepera-II robots built by K-Team, shown in Fig. 2. The robots are non-holonomic, featuring two-wheeled differential-drive. The wheels are independently controllable via calls to supplied position- or velocity-control functions, and their positions are measured by wheel encoders with $\frac{1}{12}mm$ resolution. According to Canudas de Wit's wheeled-robot definitions in [21], this is a Type (0,2) robot. The Khepera-II robot bases are nominally 69mm in diameter, and 34mm tall (from bottom of wheels to top of base circuit board). The wheels are 15mm in diameter, and the wheel separation is 52.5mm from center to center.

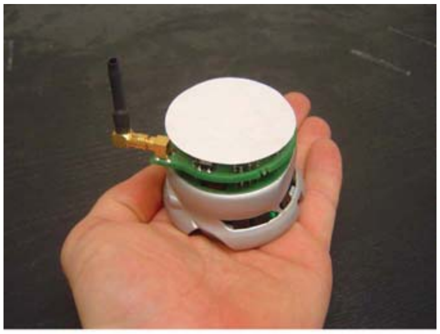


Figure 2. Khepera-II with top turret covered by white paper disc.

The robots utilize a Motorola 32-bit processor and Flash memory for storing user-created programs. Each robot has eight infrared proximity sensors, a modular expansion system allowing additional functionality, and a rechargeable battery system providing roughly one hour of operation on full charge.

In the following subsections, we provide a brief description of the robot components, including the difficulties in using them based on our experience.

Programming

Implementing custom operational programs for the robots is straightforward when utilizing the KTProject library of functions (included with the purchase of Khepera-II robots or via download from the K-Team website). The KTProject IDE includes several sample C source code files that can be quickly compiled and downloaded to the robots via the wired COM link. The robots provide a user-flash capability, allowing the user to burn an image into memory after download.

Motion Control

The library of functions provided in the KTProject IDE includes closed-loop motor control functions. The inclusion of such functions facilitates simple development and quick integration of user functions, albeit with some shortcomings.

While the supplied wheel velocity PID controller delivers fairly repeatable results, there is an apparent lack of an integrator anti-windup mechanism. Thus, if a robot encounters an object that impedes its commanded motion, the torque commands to the wheels continue to increase until either the wheel-surface friction is overcome or the robot pushes past the obstacle. In the latter case, the robot accelerates to velocities much greater than the maximum values set in the controller. This issue must be overcome with custom programming.

The position control is quite useful for commanding open-loop trajectories with simple geometry. It uses a time-optimal velocity profile, commanding the maximum acceleration until maximum velocity is achieved, followed by a coast phase, and concluding with maximum deceleration to stop at the commanded position. This time-optimal servo-control works well for single-degree-of freedom actuators. This 1-DOF control method fails for motion of a body whose kinematics couple the actuators. Since the target position is expressed as a specific number of encoder counts for each wheel, it is easy to see that the final position and orientation of the robot may vary greatly from the target position and orientation if the rate of each wheel is not monitored closely. Small errors in the control loops cause significant variations in robot angular rate, i.e., changing the orientation or heading.

Sensing Ability

Each robot is equipped with eight infrared proximity sensors. When used to detect non-absorptive and non-dissipative obstacles, the sensors perform well with a usable range of roughly 10-50mm. The minimum detection distance does not adversely affect the utility of the data, as the sensors are set back in the body of the robot roughly 10mm. The steep decay of the negative-exponential sensing function may induce large errors when using the proximity sensors for distance measurement. Additionally, the maximum detection distance may vary widely depending on the incident surface and ambient light level. In particular, flat-black surfaces prove nearly undetectable by the sensors. As such, these sensors are most effectively used to detect the *presence of* an obstacle rather than to measure the *distance to* an obstacle.

The sensor data is easily accessed through the included software library. The difficulty lies in using the data when the objects in the environment vary in reflectivity.

Communication modes

The modular expansion system of the Khepera-II robots allows the capabilities to be extended via modules called turrets, including the General Purpose I/O turret (GPIO), the Gripper Turret, the Radio Turret (RT) and the High Speed Radio Turret (HSRT). Without the radio turrets, only simple, wired two-way communication with typical PC hardware is possible. The communication method to be used is dictated largely by the specific experiment being conducted. In this research, we consider both robot-to-computer and robot-to-robot communication.

For *robot-to-robot* communication, the RT must be used, as the HSRT is incapable of direct communication between robots. This limitation of the HSRT arises due to the host-client paradigm of the Bluetooth protocol. In a decentralized control environment, a centralized communication “switchboard” would be required in order to utilize the HSRT. Additionally, systems using the HSRT are limited to seven members (again due to the Bluetooth protocol).

For *robot-to-computer* communication, there are three off-the-shelf choices. The first, cheapest, easiest, and most reliable choice is to use the direct-wired serial port. This mode of communication has proved to be reliable both for executing the built-in functions of the robots as well as for initiating a custom executable and receiving status information from the robots. For development, the wired communication is the best choice. However, the physical constraint of the wires eliminates this option in a multiple robot or high-mobility setting. The second choice is the HSRT, which allows relatively high data transmission rates (up to 115200 *baud*). The final choice is the RT, which has proven to be somewhat less than optimal. The data rate is limited to only 9600 *baud*, and the connection is very susceptible to radio frequency interference. For simple commanding of the robots, the RT has proven to be adequate, but establishing reliable communication under a custom executable has not been achieved. The RT is currently in use for all of the experiments underway at the RAMA Lab, but in the future either the HSRT or a custom alternative (such as the custom IR communication turrets developed in [19]) must be employed.

SENSORS

External position measurements are available to the central supervisor from a Cognex Insight 5400 vision system and a pair of SICK LMS400 LADAR rangefinders. The vision system is configured to return arbitrarily-ordered pairs of *x–y* position data for a pre-specified number of detected robots. To maximize the sampling frequency, we use simple blob detection based on a background contrast comparison and an area threshold. In order to provide ample contrast with the black-painted arena, identical discs of white paper are affixed to the uppermost turret of each Khepera-II robot (see Fig. 2). Doing so, however, nullifies the vision system’s ability to measure the orientation of the blobs, as there is no asymmetry. Additionally, in a multi-robot setting the robots are indistinguishable from one another, requiring a limited-information identification method. The Cognex vision system provides robust and accurate positional data, but if multiple robots are present in the field of view, the system is not able to uniquely distinguish them from one another. The issue of maintaining track of individual robots in order to properly and consistently apply the measurements is addressed below.

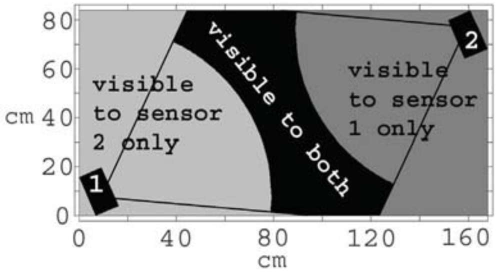


Figure 3. LADAR Coverage of the RAMA Lab Swarm Arena

Range measurements across the arena are provided by two SICK LMS400-1000 industrial LADAR Measurement Systems, communicating with the central host via a 10MBit, half-duplex Ethernet (TCP/IP) connection. The units utilize diode lasers and a rotating mirror to sweep across 70°.

The LMS400 LADAR unit allows the user to configure several parameters of the scan. Scan frequency may be set from 360Hz-500Hz, with angular resolution ranging from 0.1333°-1.0°. Measured distances (at each angular point) are reported from 500mm-3000mm. Range resolution is 1mm, although the reported systematic measurement standard error is 4mm for typical remission values of 40-100%, reaching as high as 10mm for lower remission values. Remission values below 6.5% lead to invalid measurements, reported explicitly as 0mm. To facilitate optimal data collection, range, edge, median, and mean filters may be set. In the experiments conducted, these capabilites have not been exploited; all such parameters were left at factory default values.

Two LADAR units are utilized in the RAMA Lab swarm arena for three basic reasons. The first two are the minimum range and limited detection angle of the LADAR units. Utilizing two units at opposite ends of the arena ensures complete coverage of the entire arena by at least one sensor (see Figure 3). The third reason is the susceptibility to line-of-sight obstruction; that is, one robot may occlude one (or more) other robot(s) lying within the wedge defined by the tangents from the first robot to

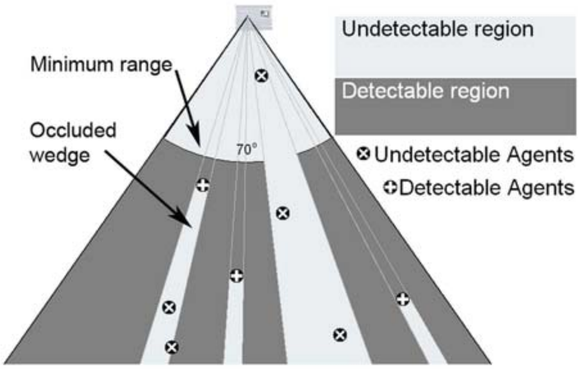


Figure 4. Partial and full occlusion of distant robots by a proximal robot. Image is approximately to scale.

the center-point of the LADAR unit (see Figure 4).

Each of the two SICK LMS400 LADAR sensors returns a set of range data covering the 70° "wedge", discretized into 0.25° samples. At each angular point, a range measurement is returned. This data is simply a sweep over the field of view of the sensor, so the positions of the robots within the field of view must be extracted. This is discussed later in the *Arc Detection* subsection of the *ESTIMATION METHODS* section.

ESTIMATION METHODS

Due to the nearly uniform cylindrical shape of the Khepera-II robots, it is not possible to directly extract the orientation from the range data returned by the LADAR sensors. The vision system has the capability of returning orientation data provided there are distinguishable features on the robot when viewed from above; however, the orientation measurements require significant additional processing time in a scalable problem such as swarm or cooperative robotics, and are thus disabled.

Reliance on visual pattern recognition for extracting orientation and identity information is used by other research groups, e.g., see [19, 22], as well as in earlier experiments by our group [23]. Robot orientation or pose is also estimated under many SLAM implementations, e.g., see [10, 15]. Our approach eliminates the need for complex machine-vision software and processing capabilities.

Algorithms have been developed to allow data collected by the vision and LADAR sensors to be used for experimental data collection and performance validation (metrology), as well as for feedback to the swarm.

Arc Detection

The range data from the SICK LADAR units is a raw set of points, so the data must be processed to extract the locations of the robots. Traditional methods for extracting features from this sort of 1-D data include wavelet transforms [24] and the Circular Hough Transform [25, 26]. Wavelets can be used quite successfully for locating, within a set of data over an independent variable, features of interest. For perceiving specific shapes, appropriate wavelets must be used.

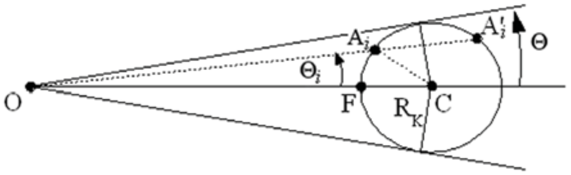


Figure 5. Geometric relationships for arc detection in LADAR data. R_K is the radius of the Khepera-II robot. The angular extent of the robot at range $\overline{OF}$ is 2Θ.

The authors have developed another method for arc detection, akin to a correlation function. With knowledge of the geometry of the Khepera-II robots, it is possible to predict the signal returned by the LADAR sensors if a robot is in its field of view. Simple trigonometry defines the angular extent (2Θ) of the robot at a given range ρ, as illustrated in Fig. 5. The formulae and algorithm for calculating the predicted arc are given in Eq.(1). The predicted arc is stored to a vector $S = \{S_{-\frac{N-1}{2}}, ..., S_0, ..., S_{\frac{N-1}{2}}\}$ of length N. The incremental angle used in the algorithm is $\Delta\Theta$, and the estimated range to the robot at angle Θ_i is S_i. Relating Fig. 5 to Eq.(1): $\rho = \overline{OF}$, $\psi_i = \angle OA_iC$, $\phi_i = \angle OCA_i$, $R_K = \overline{CA_i} = \overline{CF}$, $S_i = \overline{OA_i}$.

$$
\begin{aligned}
&\Theta = \arcsin(\frac{R_K}{\rho + R_K})\\
&N = \text{floor}(\frac{2\Theta}{\Delta\Theta})\\
&S_0 = \rho\\
&\text{if } N \text{ even, } N = N-1\\
&\text{for } i \in \{-\frac{N-1}{2}, \ldots, -1, 1, \ldots, \frac{N-1}{2}\}\\
&\quad \Theta_i = |i\Delta\Theta|\\
&\quad \psi_i = \arcsin\left(\frac{\rho + R_k}{R_k}\sin\Theta_i\right)\\
&\quad \text{if } \psi_i < \pi/2, \ \psi_i = \pi - \psi_i\\
&\quad \phi_i = \pi - \psi_i - \Theta_i\\
&\quad S_i = R_K\frac{\sin\phi_i}{\sin\Theta_i}
\end{aligned}
\tag{1}
$$

At short ranges just larger than sensor minimum range, the robots extend over approximately 5.46°, which corresponds to 21 sample points at 0.25° angular resolution. At the maximum extent of the arena, the robots exhibit an angular extent of approximately 1.8°, extending over only 7 sample points.

Using these equations, a *predicted arc* is created for every point in the raw data. The raw data surrounding the current point is truncated to the same length as the predicted arc, i.e., N in Eq.(1). These two vectors are unit-normalized, and their inner-product is evaluated. The resulting scalar is then multiplied by a confidence factor; this final product is the *score* for that range point. Empirically, we have found that N is a very good measure of confidence, giving strong confidence at short ranges (with many points to fit to the arc) and weak confidence at longer ranges (with few points to fit).

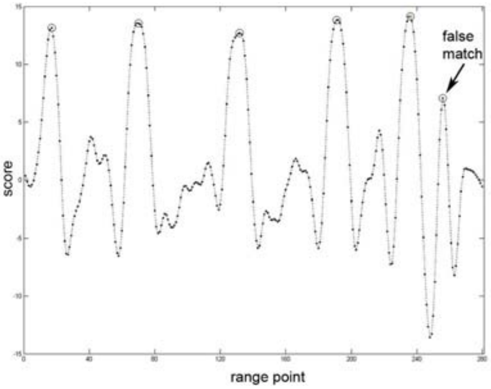


Figure 6. A sample score vector with five robots in the field of view. The right-most peak is incorrectly marked as a match since the algorithm was configured to find six robots.

Upon completion of the point-by-point scoring described above, the top scores are selected. The score data is clearly of the same length as the range data, in our case 280 points (a sample set of score data is shown in Fig. 6). The arc of a single robot clearly extends over N points (N appropriate for that range). Thus the maximum values in the score data must be separated by the appropriate number of points. When combined with a smoothing algorithm this method has proved to be, although not perfect, quite robust at identifying robots within the LADAR field of view.

This algorithm requires the number of robots to be known, so in cases where there are fewer robots in the field of view (due to range limits or occlusion as discussed above) false matches are guaranteed. The handling of these situations is discussed in the *Tracking* subsection below.

2nd Order KF

The central supervisor utilizes a 2nd-order Kalman Filter [27, 28] to maintain the four estimated states of each robot. In addition to estimating the x-y position, the filter estimates the velocity and orientation angle of each robot.

The robots are non-holonomic, thus the kinematic model can be exploited to extract the latent variable θ, the orientation angle, without knowledge of the control input. A novel addition to the typical extended Kalman Filter is the variation of the process noise intensity of state θ (the robot orientation) as a function of another state v (robot linear velocity). The state and measurement models are briefly shown below in Eqs.(2)-(3). The discrete model we use in the 2nd-order Kalman filter is defined in greater detail, and results are provided and discussed in [29].

$$\dot{X}(t) = \underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{bmatrix}}_{X(t)=[x_1\ x_2\ x_3\ x_4]^T} = \underbrace{\begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{F}(X(t))=[f_1\ f_2\ f_3\ f_4]^T} + \begin{bmatrix} 0 \\ 0 \\ \xi_v(t) \\ \xi_\theta(t) \end{bmatrix} \qquad (2)$$

$$Z(t) = \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0 \end{bmatrix}}_{C} X(t) + \begin{bmatrix} w_x \\ w_y \end{bmatrix} \qquad (3)$$

In the constant-noise model, the control variables $\dot{v}$ and $\dot{\theta}$ are modeled with zero-mean white noises $\xi_v(t)$ and $\xi_\theta(t)$, respectively. In the varying-noise model, $\xi_\theta(t)$ is treated as a function of state v, the linear velocity. This varying-noise model is discussed in depth in [23,29]. Measurements provided by the vision system and/or LADAR arc-detection algorithm are absolute position measurements (x_m and y_m) assumed to include zero-mean Gaussian-distributed noise $w_x \sim N(0,W_x)$ and $w_y \sim N(0,W_y)$. This measurement model is defined by Eq. (3).

The *prediction* is performed at each time step, initially updating states before applying the correction. These estimates are then propagated and updated in typical form. Without measurements, the system propagates only via the nonlinear state transition function. In update steps where measurements are available and valid, the *innovation* is performed, utilizing the 2nd order Taylor expansion of the dynamic model in Eq.(2). (The interested reader is directed to references [27–29] for further details.)

Tracking

When multiple robots are present in the field-of-regard for the LADAR and vision systems, the order of measurements may be rearranged from sample to sample. Additionally, there may be occasional excessive error in the measurements due to a sensor glitch or communication error, or false identifications returned either by the arc-detection described above or by the vision system. A matching and validation algorithm has been developed to detect and correct these occurrences to the extent possible.

The matching portion of the algorithm is based on the *Hungarian Algorithm*, first identified by Kuhn [30] and further developed by Munkres [31]. The algorithm solves the minimum-cost assignment problem. We treat the error between a measurement and an estimate as the cost, and the algorithm selects the order that minimizes the error.

If one or more of the measurements exhibits error that is too large due to background clutter or other error, those measurements would still be applied erroneously in the estimation algorithm. The method for data validation is adapted from Leonard's geometric beacon tracking method of localization [9].

At each prediction step k of the central supervisor's EKF, a *validation gate* g is established around the estimated output of the i-th robot, $\hat{Z}^i_{k+1} = C\hat{X}^i_k$. The range of the validation gate is dependent on the confidence in the prediction, i.e., the predicted error covariance of the estimate, S. At time-step k, the predicted state for the i-th robot is compared to the j-th measurement, Z^j_k.

$$\begin{aligned} V_{ij} &= Z^j_k - \hat{Z}^i_k \\ g^2 &\geq V_{ij} S^{-1}_{ij} V^T_{ij} \end{aligned} \tag{4}$$

Leonard refers to the size of gate g as a "number of standard deviations" of acceptable error. Bar-Shalom [32] notes instead that the confidence region defined by g is χ^2-distributed, and thus the dimension of the measurement vector must be considered. For our 2-dimensional measurement vector, a one-sided confidence region of 68.2% (roughly equivalent to one standard deviation) is given by

$$\begin{aligned} g^2 &= \chi^2_2[0.318] \approx 2.29 \\ g &\approx 1.51 \end{aligned}$$

For the matching to be successful, the Kalman Filter described above must maintain estimates within a certain error bound. Experiments and simulations at the RAMA Lab continue in the search for the appropriate relationships between acceptable error within the KF estimate and the appropriate value for g that is not too restrictive. The lowest value found to work with the real data, described below, which provides rejection of bad data but allows tracking is $g \approx 2$, corresponding to a confidence region of 86.5%.

Metrology

In a swarm experiment, external observation of the entire swarm is desirable to quantify the performance of the algorithm as well as to verify the completion of the task. For purposes such as this, it is especially beneficial to use a sensor (or set of sensors) whose observational range extends over the entire region of interest such that all robots may be observed simultaneously.

Under these considerations, the vision system implemented here is ideal. Its field of view extends beyond the bounds of the swarm arena, and none of the robots is ever occluded. Additionally, the camera provides accurate position measurement ($\sigma \approx 0.03mm$) at reasonable sampling rate (≈ 1.5Hz for 6 robots).

The major limitation of the vision system is in the scalability: the more robots are present in the field, the longer the sample period becomes as the system processes the images. The use of blob detection ensures that the sample period is minimized.

The LADAR system offers a trade-off, providing a higher sample rate that is independent of the number of robots in the field as long as the communication path is not a bottleneck. As this data is processed off-sensor, the sensor sample rate is maximized. The sacrifices to achieve this data rate are the limited and/or occluded field of view, requiring two (or more) sensors to be used for full arena coverage, and the increased sensor noise ($4mm$ standard error versus $0.03mm$ for the vision system).

Feedback

In the feedback setting, the timeliness of data samples and associated processing time becomes critical. We are currently performing experiments in which position data from the external sensors is provided to the robots at a reduced data rate. Using an odometry calibration method based on Larsen's and Martinoli's augmented Kalman Filter method [3, 5], simulations have shown that the robots require very low update rates to maintain localization accuracy. Experiments are underway to verify the performance in our real environment.

RESULTS

Data was collected from the Cognex InSight 5400 vision system with a single robot moving under the Braitenberg obstacle avoidance algorithm built in to the Khepera-II standard library. The implementation is a variant of Braitenberg's Vehicle 3b, called *Like* [33]. The resulting trajectories are semi-deterministic and depend strongly on the initial pose relative to the arena geometry as well as the variation among the onboard IR proximity sensors. These data sets were gathered using a distinct identifier disc on the robot displaying a spatial pattern such that the vision system can make direct measurements of orientation. This was desired to have a *truth* measurement for comparison with the estimated angle from the modified 2nd order Kalman Filter. The trajectories are plotted in Figure 7. The figures exclude measurements that are invalid due to misidentification or failed pattern-matching. Those bad points are identified in the algorithm and are treated as 'skipped' measurements.

The orientation of the robots in these trajectories was estimated under the model described above using both a state-varying noise model and a constant noise model. The results are given in Table 1.

	constant	varying	%better	(worse)
Set 1	0.1382	0.1475		(6.75)
Set 2	0.2212	0.2030	8.23	
Set 4	0.1276	0.1389		(8.85)
Set 5	0.2155	0.2005	7.00	

Table 1. Estimation scores (RMS errors) for the constant-noise and varying-noise models, expressed in radians.

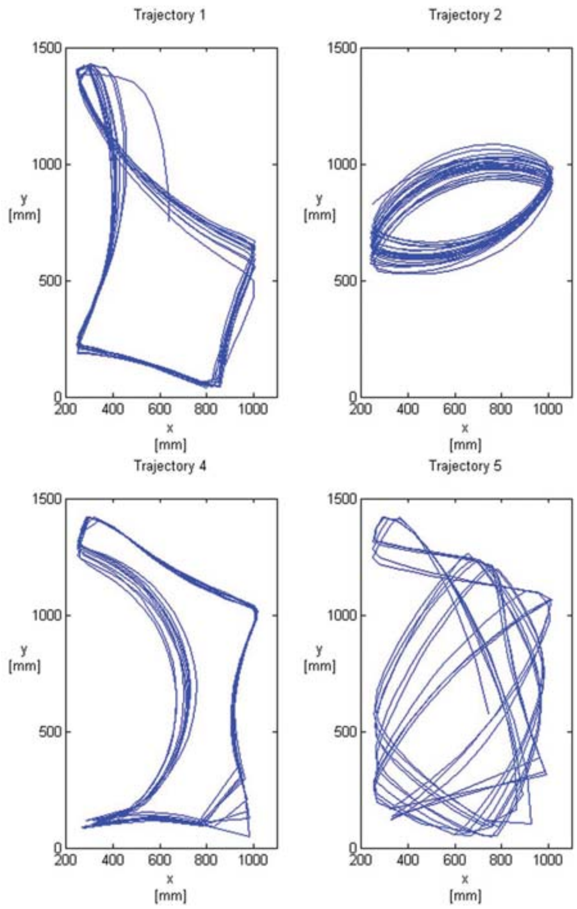


Figure 7. Four of the five measured trajectories.

The relative performance of the varying-noise model compared to the constant-noise model was evaluated by comparing the mean squared error between the measured and estimated angle, i.e., $\sqrt{\sum_k (\mathrm{mod}(\theta_k - \hat{\theta}_k, 2\pi))^2}$. The modulo function is critical to account for angle-wrapping, i.e., the fact that $\hat{\theta} = \hat{\theta} + 2\pi$.

CONCLUSION

A functional arena for experiments in swarm robotics has been established. Sensing and estimation capabilities have been implemented, providing good metrology results as well as the ability to provide online feedback to swarm members. Preliminary experiments have shown that this testbed will provide a good basis for future swarm research and experiments at the RAMA Lab.

Future development plans include improved communication ability. Under consideration is the use of both RTs and HSRTs, for robot-robot and robot-supervisor communication respectively. Alternatively, GumStix small-form-factor computers may be used alongside the HSRT (using Bluetooth communication) to allow robot-to-robot communication. This option allows the inclusion in our facility of other robots with Bluetooth communication ability, such as the iRobot Create.

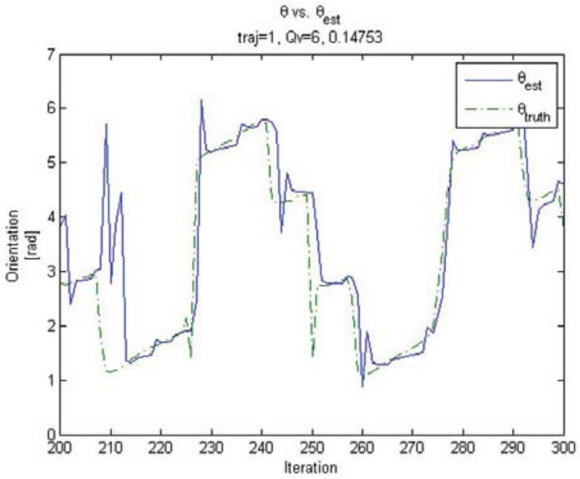


Figure 8. A sub-sample of trajectory set 1 comparing θ_{truth} to $\hat{\theta}$ for the varying-noise model.

This testbed and the estimation methods we have developed present us with opportunities for real-world validation. The estimation algorithms have been developed to be independent of the specific sensors. The arc-detection method may be applied to other sets of range-data and other shapes to be detected. The orientation estimation may be applied to any sensor that provides 2D position data (e.g., vision, GPS, RFID tag/reader, ultrasonic, etc.), maximizing our choices.

ACKNOWLEDGMENT

This work was supported by the Army Research Office under grant number W911NF-08-1-0106.

REFERENCES

[1] Borenstein, J., and Feng, L., 1996. “Measurement and correction of systematic odometry errors in mobile robots”. *Robotics and Automation, IEEE Transactions on,* **12**(6), Dec, pp. 869–880.

[2] Chong, K. S., and Kleeman, L., 1997. “Accurate odometry and error modelling for a mobile robot”. *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on,* **4**, Apr, pp. 2783–2788.

[3] Larsen, T. D., Bak, M., Andersen, N. A., and Ravn, O., 1998. “Location estimation for an autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry”. *FUSION98 SPIE Conference, Las Vegas*, Jul.

[4] Von der Hardt, H.-J., Husson, R., and Wolf, D., 1998. “An automatic calibration method for a multisensor system: application to a mobile robot localization system”. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on,* **4**, May, pp. 3141–3146.

[5] Martinelli, A., Tomatis, N., Tapus, A., and Siegwart, R., 2003. "Simultaneous localization and odometry calibration for mobile robot". *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on,* **2**, Oct., pp. 1499–1504.

[6] Antonelli, G., and Chiaverini, S., 2006. "Linear estimation of the odometric parameters for differential-drive mobile robots". *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct., pp. 3287–3292.

[7] Murata, S., and Hirose, T., 1993. "Onboard locating system using real-time image processing for a self-navigating vehicle". *Industrial Electronics, IEEE Transactions on,* **40**(1), Feb, pp. 145–154.

[8] Kato, K., Ishiguro, H., and Barth, M., 1999. "Identifying and localizing robots in a multi-robot system environment". *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on,* **2**, pp. 966–971.

[9] Leonard, J., and Durrant-Whyte, H., 1991. "Mobile robot localization by tracking geometric beacons". *Robotics and Automation, IEEE Transactions on,* **7**(3), Jun, pp. 376–382.

[10] Lin, H.-H., Tsai, C.-C., and Hsu, J.-C., 2008. "Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering". *Instrumentation and Measurement, IEEE Transactions on,* **57**(9), Sept., pp. 2024–2034.

[11] Rekleitis, I., Dudek, G., and Milios, E., 1997. "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error". *International Joint Conference in Artificial Intelligence (IJCAI)*.

[12] Roumeliotis, S., and Bekey, G., 2002. "Distributed multi-robot localization". *Robotics and Automation, IEEE Transactions on,* **18**(5), Oct, pp. 781–795.

[13] Thrun, S., 2008. *Simultaneous Localization and Mapping*. Springer Berlin / Heidelberg, pp. 13–41.

[14] Lu, F., and Milios, E., 1997. "Robot pose estimation in unknown environments by matching 2D range scans". *Journal of Intelligent and Robotic Systems,* **18**(3), pp. 249–275.

[15] Stachniss, C., 2006. "Exploration and mapping with mobile robots". PhD thesis, University of Freiburg, Department of Computer Science, April.

[16] Se, S., Lowe, D., and Little, J., 2005. "Vision-based global localization and mapping for mobile robots". *Robotics, IEEE Transactions on,* **21**(3), June, pp. 364–375.

[17] Hahnel, D., Burgard, W., Fox, D., Fishkin, K., and Philipose, M., 2004. "Mapping and localization with RFID technology". In Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, Vol. 1, pp. 1015–1020.

[18] Deyle, T., Kemp, C., and Reynolds, M., 2008. "Probabilistic UHF RFID tag pose estimation with multiple antennas and a multipath RF propagation model". *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept., pp. 1379–1384.

[19] Hoyt, S., McKennoch, S., and Bushnell, L. G., 2005. "An autonomous multi-agent testbed using infrared wireless communication and localization". *UWEE Technical Report Number UWEETR-2005-0005*.

[20] Holland, O., Woods, J., De Nardi, R., and Clark, A., 2005. "Beyond swarm intelligence: the UltraSwarm". *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, June, pp. 217–224.

[21] Canudas de Wit, C., 1996. *Theory of Robot Control*. Spring-Verlag.

[22] Jung, D., Heinzmann, J., and Zelinksy, A., 1998. "Range and pose estimation for visual servoing of a mobile robot". *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on,* **2**, May, pp. 1226–1231.

[23] Fricke, G. K., 2009. "Localization, tracking, and odometry calibration of a multi-agent swarm system". Master's thesis, Duke University, Department of Mechanical Engineering and Materials Science, Durham, NC, April.

[24] Daubechies, I., 1990. "The wavelet transform, time-frequency localization and signal analysis". *Information Theory, IEEE Transactions on,* **36**(5), Sep, pp. 961–1005.

[25] Hough, P. V. C., December 18, 1962. Method and means for recognising complex patterns. U.S. Patent 3069654.

[26] Duda, R. O., and Hart, P. E., 1972. "Use of the Hough transformation to detect lines and curves in pictures". *Commun. ACM,* **15**(1), pp. 11–15.

[27] Kalman, R. E., 1960. "A new approach to linear filtering and prediction problems". *Journal of Basic Engineering, ASME,* **82**, pp. 35–45.

[28] Athans, M., Wishner, R., and Bertolini, A., 1968. "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements". *Automatic Control, IEEE Transactions on,* **13**(5), Oct, pp. 504–514.

[29] Fricke, G. K., Milutinović, D., and Garg, D. P., 2009. Robotic pose estimation via an adaptive Kalman filter using state-varying noise. Submitted to *IASTED Robotics and Applications Conference*, Cambridge, MA, Nov 4-6, 2009.

[30] Kuhn, H. W., 1955. "The Hungarian method for the assignment problem". *Naval Research Logistics Quarterly,* **2**(1-2), pp. 83–97.

[31] Munkres, J., 1957. "Algorithms for the assignment and transportation problems". *Journal of the Society for Industrial and Applied Mathematics,* **5**(1), pp. 32–38.

[32] Y. Bar-Shalom, T. E. F., 1988. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA.

[33] Braitenberg, V., 1984. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, Massachusetts.

A.2.5 Robotic Pose Estimation via an Adaptive Kalman Filter using State-Varying Noise

The following paper (on the next 7 pages) was co-authored by Gregory K. Fricke, Dejan Lj. Milutinović, and Devendra P. Garg, and appeared in the Proceedings of the 2009 IASTED Robotics and Automation Conference, published and presented in November 2009.

Proceedings of the 14th IASTED International Conference
Robotics and Applications (RA 2009)
November 2 - 4, 2009 Cambridge, MA, USA

ROBOTIC POSE ESTIMATION VIA AN ADAPTIVE KALMAN FILTER USING STATE-VARYING NOISE

Gregory K. Fricke
Mechanical Engineering
& Materials Science
Duke University
Box 90300, Hudson Hall 144
Durham, NC 27708
United States of America
email: gregory.fricke@duke.edu

Dejan Milutinović
Applied Mathematics
& Statistics
University of California at Santa Cruz
Mail Stop SOE2, 1156 High Street
Santa Cruz, CA 95064
United States of America
email: dejan@soe.ucsc.edu

Devendra P. Garg
Mechanical Engineering
& Materials Science
Duke University
Box 90300, Hudson Hall 144
Durham, NC 27708
United States of America
email: dpgarg@duke.edu

ABSTRACT
Swarm robotics offers the promise of enhanced performance and robustness relative to that of individual robots, with decreased cost or time-to-completion for certain tasks. Having many degrees of freedom, the swarm related control and estimation problems are quite challenging, specifically when the solutions involve a large amount of communication among the robots. Under certain sensing modalities, direct measurement of robot orientation may either be not possible or require excessive sensor processing. In this paper, a novel method is presented to vary process noise intensity as a function of an estimated state in order to arrive at the hidden state robot orientation. Experimental results are provided, demonstrating the efficacy of the method as well as the error reduction relative to fixed-noise estimation.

KEY WORDS
Collective robotics, Pose estimation, Kalman filter, Stochastic estimation, Mobile robotics, Differential-drive

1 Introduction

Robotic pose estimation is critical for proper feedback control of robot trajectory. Pose estimation and robot localization have been studied extensively, utilizing methods such as dead-reckoning [1–6], on-board and off-board computer vision [7, 8], environmental sensing [9–11], collective or distributed localization [12, 13], and simultaneous localization and mapping (SLAM) [14] using laser-scanning systems [15, 16], on-board cameras [17], or ultrasonic rangefinders [11]. Most of the above works rely on some variant of the Kalman Filter [18] or Extended Kalman Filter [19].

Dead-reckoning solutions are notoriously susceptible to accumulated error due to modeling errors or unmodeled dynamics such as wheel-slippage or uneven surfaces. The above references include several methods for odometry correction or calibration, all of which require some secondary form of sensing for feedback correction.

Computer vision-based localization systems generally come in two flavors: off-board and on-board. Off-board implementations feature either high-speed sensors with high-complexity pattern recognition and tracking software (requiring extensive computational capability and advanced machine-vision techniques) or simpler vision systems reliant on known configurations or patterns on the robot(s) being tracked (see, e.g., [20, 21]).

On-board solutions rely on feature tracking in the environment, requiring that the environment contains sufficient features to identify and track. Early work required *a priori* known environmental maps or beacon locations [7, 9]; more recent SLAM methods build the environmental map as the robot moves [14, 16]. Similar environmental constraints are placed on laser-scanning- and ultrasonic rangefinder-based SLAM implementations. Additionally, on-board solutions necessarily place significantly higher sensing and computation requirements on the individual robots.

1.1 Problem formulation

This research group, performing research in collective robotics and decentralized formation control, seeks to perform algorithmic development, test, and evaluation in the presence of real-world noise with minimal equipment overhead. As a surrogate for significantly more complex sensing systems on-board the robots, measurements of the system will be made by external position sensors and made available to the individual robots via direct or broadcast wireless communication. Such global information sharing may be considered analogous to the information sharing utilized in [22]. If, for a particular experiment, sensing or communication range is assumed to be limited for individual robots, direct central-computer-to-robot communication could be configured to only report information that the robot would accordingly have access to if the sensing ability were on-board. This type of experimental environment may be called *mixed-simulation testing*, indicating a combination of real and simulated components. Such testbeds are used extensively for validation of complex systems such as spacecraft [23].

The robotic agents forming the basis of the preliminary testbed development are Khepera-II differential drive robots from K-Team. External observations of robots in the testbed are provided by two SICK LMS400 LADAR sensors and an overhead Cognex Insight 5400 industrial machine-vision system. The LADAR sensors provide measurement of the $x - y$ position of multiple agents within the swarm arena. However, due to the nearly uniform cylindrical shape of the Khepera-II robots when viewed from the side, it is not possible to directly measure the agent orientation θ from the range data returned by the LADAR sensors. The vision system has the capability of returning orientation data *provided there are distinguishable features which correlate with the robot orientation* when viewed from above. Measuring the robot orientation in this way requires significant additional processing time, especially in the case of scalable problems related to swarm and cooperative robotics.

In this paper we present an alternative approach in which the orientation of the agent is treated as a hidden variable, and is estimated through observation of the robot motion. The dynamic model of the robot (specifically, the non-holonomicity) is then exploited to appropriately vary the process noise covariance used in the extended Kalman Filter (EKF). Recent papers have discussed covariance variation via fuzzy logic and with dependence on time-step k [11, 24]; the authors of this paper are not aware of prior work utilizing state estimates to vary the covariance.

2 Model Definition

The pose of an individual agent in a swarm is defined by the coordinates $[x, y, \theta]^T$, where x and y are the agent coordinates in a rectangular reference frame, and θ is the orientation angle with $\theta = 0$ lying on the x-axis. The agent posture changes via the agent's on-board control loop, which defines the agent's linear acceleration $\dot{v}$ and angular velocity $\dot{\theta}$. The dynamic model of the robot used in this paper is given in state space form in Eqs.(1-3).

$$X(t) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ \theta(t) \end{bmatrix} \tag{1}$$

$$\mathbf{F}(X(t)) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

$$\dot{X}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{bmatrix} = \mathbf{F}(X(t)) + \begin{bmatrix} 0 \\ 0 \\ \xi_v(t) \\ \xi_\theta(t) \end{bmatrix} \tag{3}$$

Knowledge of these control inputs $\dot{v}$ and $\dot{\theta}$ would greatly decrease the error in the estimation. In many cases, such as the one described in this paper for an external observer, the control inputs are unknown. For this reason, the control variables $\dot{v}$ and $\dot{\theta}$ are modeled with the zero-mean white noises $\xi_v(t)$ and $\xi_\theta(t)$, respectively. Measurements provided by the vision system and/or LADAR pair are absolute position measurements x_m and y_m assumed to include zero-mean Gaussian-distributed noise $w_x \sim N(0, W_x)$ and $w_y \sim N(0, W_y)$. This measurement model is defined in Eq.(4).

$$\begin{aligned} Z(t) &= \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{C} X(t) + \begin{bmatrix} w_x \\ w_y \end{bmatrix} \end{aligned} \tag{4}$$

We utilize stochastic signals as a general model of the unknown control signals. The advantage of modeling the unknown control signals as stochastic signals is that we can exploit Kalman filter theory to estimate unknown speed v and angle θ based only on the position measurements x_m and y_m.

We initially assume that the white noise signals that model the control variables are uncorrelated. To establish the baseline, we also assume that they are of constant intensity. In the rest of this paper we will consider the posture estimation based on this constant intensity model as compared to another model in which the noise intensity ξ_θ varies as a function of estimated robot velocity.

The rationale behind the velocity-varying models for estimation lies in the intuition that the largest changes in orientation are likely to be experienced when the velocity v is small. This is due to the observation that for a differential-drive robot, fast orientation changes result from large differences in wheel speeds. For such a system with a limited dynamic range of wheel speeds, the maximum rate of orientation change will occur when the two wheels are each driven at the maximum wheel speed but in opposite directions, resulting in zero linear velocity. At the other extreme, maximum linear velocity is achieved when both wheels are driven at maximum speed in the same direction, resulting in zero angular velocity. Additionally from an observational standpoint, in the limit as velocity goes to zero, changes in x_m and y_m will be zero or undetectable while the orientation may change in range $[0, 2\pi]$.

2.1 Orientation Estimation

Our proposed method is based on the 2nd-order Kalman filter [19] for non-linear systems. Individual agents in the swarm are characterized by the state vector $X = [x, y, v, \theta]^T$, and the continuous state space model is defined by Eqs.(1-3). The Kalman filter is based on two steps: the prediction and the update. The prediction and update step equations are presented below without derivation. Additional details are available in references [18, 19].

In the *prediction step*, the computation of the predicted value $\bar{X}$ of the state vector and its covariance are based on the last estimation $\hat{X}$. Since we are dealing with data that is regularly sampled with sampling period Δt, we use the discrete time version of the 2nd-order filter. The subscript k indicates the iteration step; the predicted state vector at the $k+1$ sample is given by

$$\begin{aligned} \bar{X}_{k+1} &= E_{\hat{X}_k,P}\left\{X_{k+1}\right\} \\ &= E_{\hat{X}_k,P}\left\{X_k + \mathbf{F}(X_k)\Delta t\right\} \end{aligned} \tag{5}$$

where the expectation is in regard to Gaussian distribution with the mean value $\hat{X}_k$ and corresponding covariance P. Using the Taylor Series expansion up to the second-order terms and including $E_{\hat{X}_k,P}(X_k - \hat{X}_k) = \mathbf{0}$ the above expression can be rewritten as

$$\begin{aligned} \bar{X}_{k+1} = \hat{X}_k + \mathbf{F}(\hat{X}_k)\Delta t + \\ \frac{1}{2}E\left\{\sum_{i=1}^{4}\phi_i(X_k - \hat{X}_k)^T F_i(X_k - \hat{X}_k)\right\} \end{aligned} \tag{6}$$

in which the vectors ϕ_i are: $\phi_1 = [1\ 0\ 0\ 0]^T$, $\phi_2 = [0\ 1\ 0\ 0]^T$, $\phi_3 = [0\ 0\ 1\ 0]^T$, $\phi_4 = [0\ 0\ 0\ 1]^T$ (i.e., typical orthonormal basis vectors of $\Re^4$) and the matrix F_i is

$$[F_i]_{mn} = \frac{\partial^2 f_i(\hat{X}_k)}{\partial x_m \partial x_n}\Delta t,\ m,n = 1,2,3,4. \tag{7}$$

The notation $[\ \cdot\ ]_{mn}$ denotes the element in row m and column n. We can then rewrite Eq.(6) as shown in Eq.(8).

$$\bar{X}_{k+1} = \hat{X}_k + \mathbf{F}(\hat{X}_k)\Delta t + \frac{1}{2}\sum_{i=1}^{4}\phi_i tr\left\{F_i P_k\right\} \tag{8}$$

The square matrix P_k denotes the covariance matrix of the state estimate, i.e., $P_k = E\left\{(X_k - \hat{X}_k)^T(X_k - \hat{X}_k)\right\}$. From Eq.(2) and Eq.(7), we obtain $F_3 = F_4 = \underline{0}$, and F_1 and F_2 as defined in Eqs.(9-10).

$$F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\Delta t\sin(\theta_k) \\ 0 & 0 & -\Delta t\sin(\theta_k) & -\Delta t v_k\cos(\theta_k) \end{bmatrix} \tag{9}$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta t\cos(\theta_k) \\ 0 & 0 & \Delta t\cos(\theta_k) & -\Delta t v_k\sin(\theta_k) \end{bmatrix} \tag{10}$$

Based on P_k we can also predict the covariance matrix of the state estimate $\bar{P}_{k+1}$ via Eq.(11).

$$\bar{P}_{k+1} = A_k P_k A_k^T + Q_k \tag{11}$$

The state-transition matrix A_k is defined in Eq.(12).

$$A_k = \begin{bmatrix} 1 & 0 & \Delta t\cos(\theta_k) & -v_k\Delta t\sin\theta_k \\ 0 & 1 & \Delta t\sin(\theta_k) & v_k\Delta t\cos\theta_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$Q_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \Xi^v & 0 \\ 0 & 0 & 0 & \Xi_k^\theta \end{bmatrix} \tag{13}$$

The process-noise covariance matrix Q_k consists of the velocity increment $(v_{k+1} - v_k) \sim N(0, \Xi_k^v)$ and the increment of the angle $(\theta_{k+1} - \theta_k) \sim N(0, \Xi_k^\theta)$. By including the control inputs for v and θ as noise, the magnitudes of Ξ_k^v and Ξ_k^θ are likely larger than would otherwise be required, but with the benefit of a method that is independent of communication with the robot.

In *update steps* (performed when measurements are available) we correct the predicted state estimation based on the measurements. The update step contains the following computations.

$$\begin{aligned} S_{k+1} &= C\bar{P}_{k+1}C^T + R \\ K_{k+1} &= \bar{P}_{k+1}C^T S_{k+1}^{-1} \\ \hat{X}_{k+1} &= \bar{X}_{k+1} + K_{k+1}(Z_{k+1} - C\bar{X}_{k+1}) \\ P_{k+1} &= (I - K_{k+1}C)\bar{P}_k \end{aligned} \tag{14}$$

The covariance matrix of the measurement noise, R, is given below.

$$R = \begin{bmatrix} W_x & 0 \\ 0 & W_y \end{bmatrix} \tag{15}$$

2.2 Variation of Orientation Covariance

As discussed above, it is likely that increments of θ will be larger as $v(t) \to 0$. For this reason instead of only considering variance Ξ_k^θ constant, we also consider the variance as a function of velocity v, i.e., $\Xi_k^\theta = \Xi_k^\theta(v_k)$.

In Eq.(11) and Eq.(13) above, the process noise covariance matrix Q_k, and more specifically Ξ_k^θ, is noted to vary from step to step, and as such must be re-evaluated after each update. Here we denote the dependence of $[Q]_{44}$ on v, such that it achieves its maximum value for $v = 0$ and decreases monotonically to $[Q]_{44} = 0$ at $v = \bar{v}$. The proposed function defining the relationship between Ξ_k^θ and v, given in Eq.(16), is based on the observed empirical histogrammatic correlation between v and $\Delta\theta$ over several data sets.

$$\Xi_k^\theta(v_k) = \arccos^2[(\frac{v_k}{\bar{v}})^4] \tag{16}$$

This function requires one tuning parameter, $\bar{v}$. After performing several optimization iterations using the Matlab®Optimization Toolbox, we observed that the normalization parameter correlates well with the histogrammatic peak of v for a given trajectory. It is clear, however, that for such a definition of $\bar{v}$, v_k will almost certainly exceed $\bar{v}$ at times, in which case the arccos function returns non-real values. Protections have thus been implemented in the software to ensure real values. The practical implication is that $|v| \geq \bar{v} \to \Xi_k^\theta(v_k) \equiv 0$.

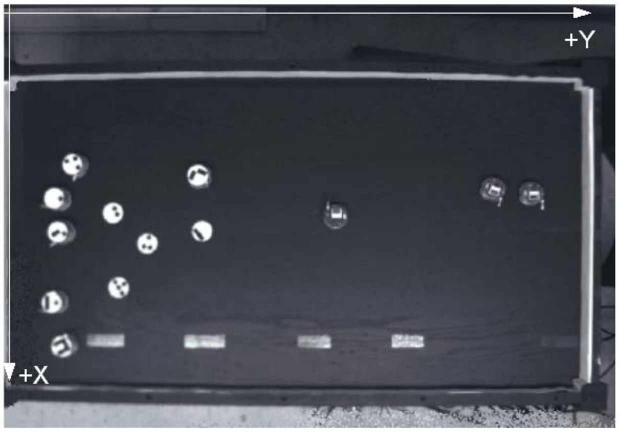


Figure 1: Overhead view of Khepera-II arena. Image taken from Cognex Insight 5400 machine vision sensor, showing several Khepera-II robots with unique identifier IDs (for truth measurements).

3 Experimental Data

The method of orientation estimation described in this paper has been implemented in the Khepera-II testbed within our experimental laboratory [25]. The testbed, shown in Figure 1 and Figure 6 has been constructed for the purpose of studying and validating swarm- and cooperative-robotics control methodologies in the presence of true sensor noise, and communication uncertainties and limitations.

Experimental data sets with the Khepera-II robots, SICK LMS-400 LADAR sensors, and the Cognex InSight 5400 vision system have been collected to provide real inputs to the algorithms. The data thus includes real-world corruption such as dropped or misread measurement packets, non-real values, noise due to environmental conditions, and sensor noise.

Data was gathered via the Cognex InSight 5400 vision system with a single robot present within the arena, moving under the Braitenberg obstacle avoidance algorithm built in to the Khepera-II standard library. The implementation is a variant of Braitenberg's Vehicle 3b, also called *Like* [26]. The resulting trajectories are non-deterministic and depend strongly on the initial orientation relative to the arena geometry, as well as on the variation among the on-board IR proximity sensors. Five data sets were collected with the robot moving in this manner. One of these sets was chosen arbitrarily to tune the parameter $\bar{v}$. The training trajectory is shown in Figure 2. The remaining trajectories are plotted in Figure 3.

These initial data sets were gathered using distinct identifier discs atop the robots (as visible in Figure 1) such that direct measurement of robot orientation would be possible and available for comparison. These sets were used to validate methods without the concerns of processing time for the vision system. These data sets contain only data from the Cognex vision system. Four additional data sets were taken with a stationary robot in different locations to quantify the sensor noise covariances (i.e., measurement noise R), including evaluation of whether the sensor error is a function of location within the arena. For the Cognex vision system, location does not significantly affect the noise covariance; the SICK LADAR sensor noise does vary significantly across the field of view. Results are given below for the data collected by the Cognex vision system.

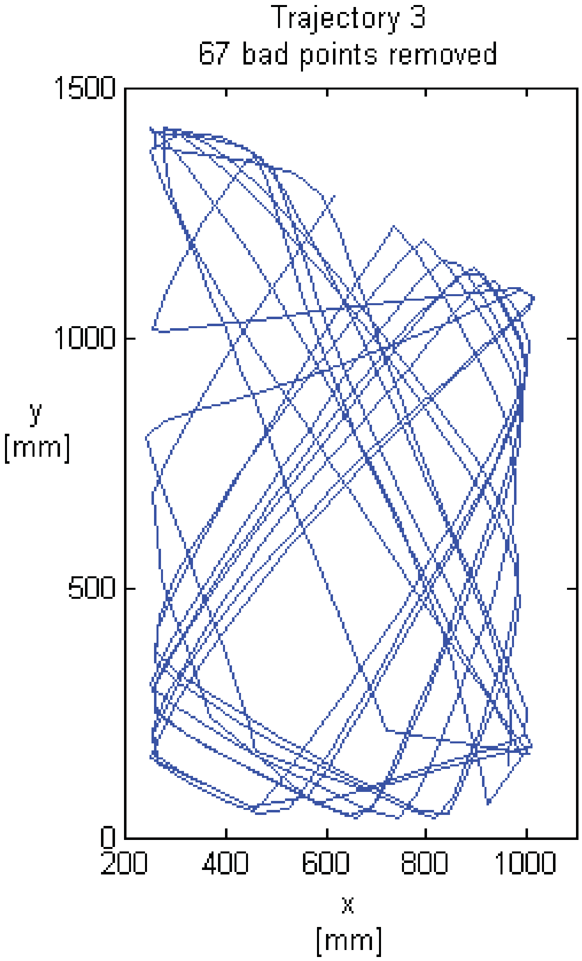


Figure 2: The robot trajectory used to find the optimal value of $\bar{v}$. The figure excludes invalid measurements, which are identified in the algorithm and treated as 'skipped' measurements.

4 Experimental Results

Simulations conducted with the above described data sets as inputs allowed the tuning of model parameters, as the estimated states could be directly compared to the measured states. The relationship between the noise intensity Ξ_k^θ and estimated velocity proposed in Eq.(16) has been optimized. The tuning parameter for this velocity-dependent noise profile was optimized under an arbitrarily chosen trajectory set from the single-robot data. For a fair and legitimate com-

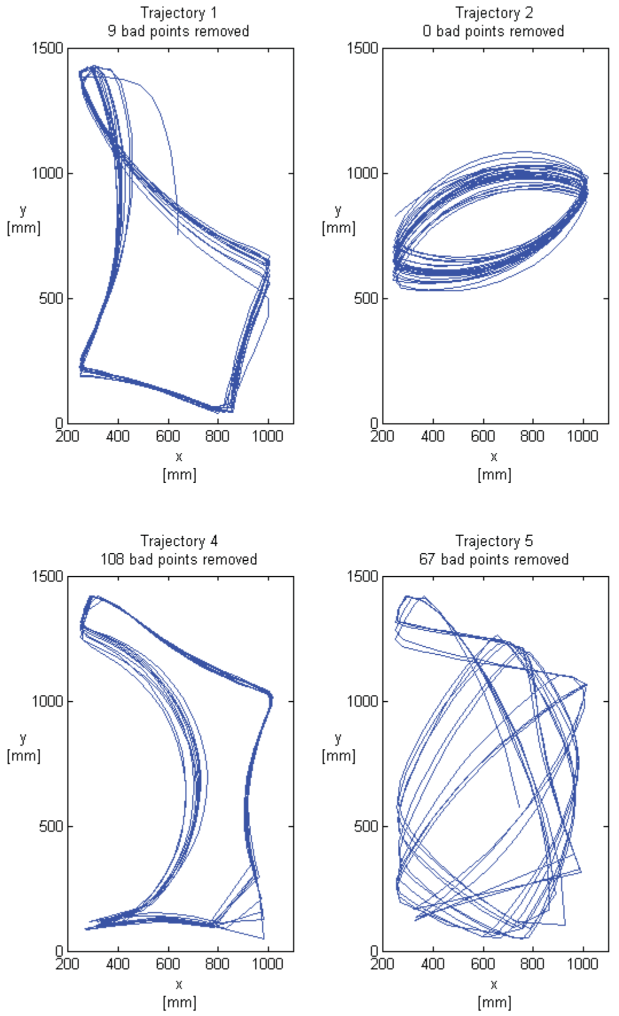


Figure 3: Four of the five measured trajectories. The figures exclude invalid measurements, which are identified in the algorithm and are treated as 'skipped' measurements.

parison, the fixed-noise covariance magnitude was also optimized under the same trajectory. The varying-noise function is plotted in Figure 4, with the fixed-noise value also plotted for comparison. As seen in the plot, the optimized parameter is $\bar{v} = 756\text{mm/s}$.

The relative performance of the varying-noise model compared to the constant-noise model was evaluated by comparing the mean squared error between the measured and estimated angle, i.e., $\sqrt{\sum_k (\text{mod}(\theta - \hat{\theta}, 2\pi))^2}$. The modulo function is critical to account for angle-wrapping; i.e., to account for the fact that $\hat{\theta} = \hat{\theta} + 2\pi$. Table 1 gives the scores (RMS error, expressed in radians) for the constant-noise and variable-noise models for each of five trajectories. The greatest score improvement (error reduction) is for Set 3, which was used as the training set for optimizing $\bar{v}$. Figure 5 illustrates the difference between measured and estimated orientation. The measured data is real and includes noise, evidenced by some non-smooth segments.

Table 1
Estimation error

	constant	varying	%change
Set 1	0.1382	0.1475	6.75
Set 2	0.2212	0.2030	-8.23
Set 3*	0.2168	0.1881	-13.24
Set 4	0.1276	0.1389	8.85
Set 5	0.2155	0.2005	-7.00

Relative performance (RMS angular error, radians) of the varying-noise model compared to the constant-noise model.

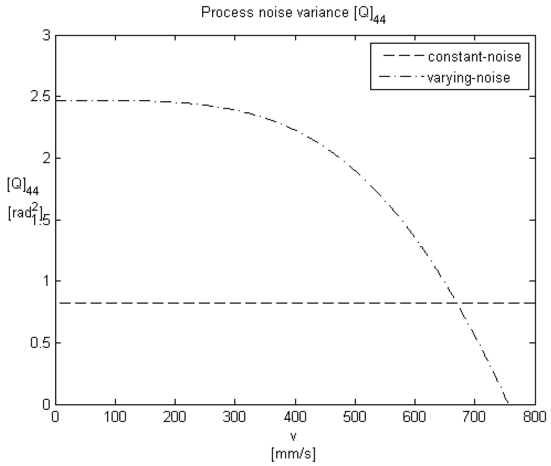


Figure 4: $\Xi_k^\theta(v)$ candidate function.

5 Applicability

This methodology can be applied on similar, differential-driven robot system utilizing position-only feedback. Use of this method has the potential to greatly reduce the sensing requirements in situations where orientation information is desired. For example, for beacon-, *a-priori*-known map-, or time-of-flight-based navigation systems could reduce the number of beacons, features, transmitters, etc., such that only position measurements are required.

In this paper, it is asserted that this estimation method is possible due to the physical constraints of a differential-drive vehicle. However, experiments have been performed only with a single trajectory control method, and thus this claim remains unproven. Continuing experiments and simulations seek to answer the question of whether this method is applicable under other trajectory control paradigms.

These experiments have focused solely on estimation via an external observer, alleviating high-throughput communication of control input by the robot to the observer. If this method is implemented on-board the robot, utilizing its own exteroceptive sensing ability to yield x-y measurements, the control inputs would be available to the estimation algorithm as well, thus significantly reducing the required magnitude of the process noise injected into the

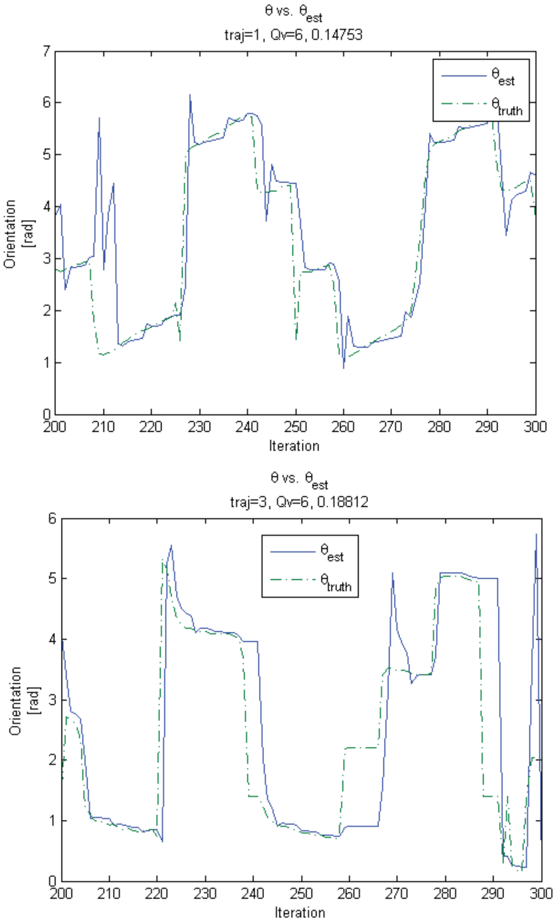


Figure 5: θ_{truth} vs. $\hat{\theta}$ for varying-noise model, trajectory sets 1 and 3. Zoomed in to show detail.

model.

This extension of the method allows the potential of greater scalability with multiple vehicle systems as the sensing and processing requirements are drastically reduced. Furthermore, applications in holonomic estimation and control may be possible. For example, a quadrotor aerial vehicle with 6DOF could exploit a system such as this *if the control input is known*.

6 Conclusion

The method of varying the estimated process noise covariance of the orientation state as a function of velocity proved to work admirably. The errors observed under the varying-noise model were, for three of the five test trajectories, significantly lower than for the constant-noise model. Clearly though, two of the trajectories result in increased estimation error with the varying-noise model. These results are compelling, and suggest that such a varying-noise model, appropriately configured, may be able to reduce estimation error. We continue to study the differences in the trajectories in an attempt to identify cases where use of a noise-varying model will be most beneficial. We suspect that the dynamic ranges of the states are critical to the convergence of the estimator; a potential method of evaluating the dynamic ranges of the trajectories has been inspired by the numerical conditioning suggested by Antonelli [6].

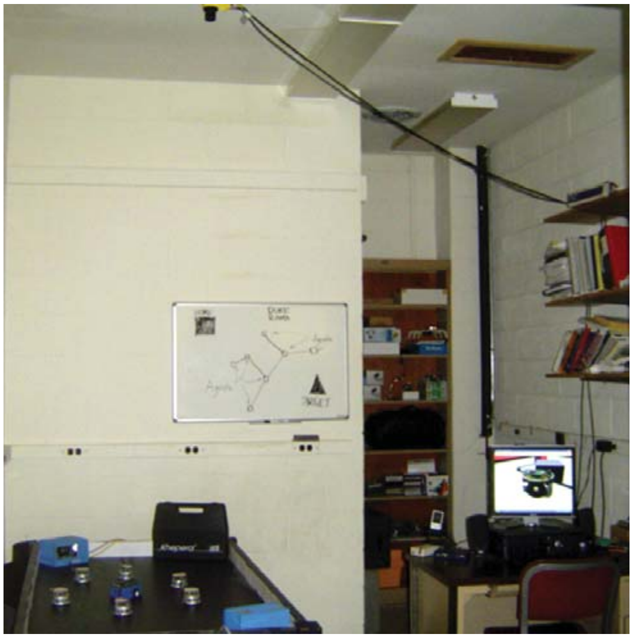


Figure 6: View of the Khepera-II lab space, showing relative locations of Cognex Insight 5400 camera and dual SICK LMS400 ladar sensors.

Acknowledgement

This work was supported by the Army Research Office under grant number W911NF-08-1-0106 titled “Modeling, Analysis, and Control of Swarming Agents in a Probabilistic Framework”.

References

[1] J. Borenstein & L. Feng, Measurement and correction of systematic odometry errors in mobile robots, *IEEE Trans. on Robotics and Automation, 12*(6), 1996, 869–880.

[2] K. S. Chong & L. Kleeman, Accurate odometry and error modelling for a mobile robot. *Proc. IEEE International Conf. on Robotics and Automation (ICRA’97)*, Apr 1997, 2783–2788.

[3] T. D. Larsen, M. Bak, N. A. Andersen, & O. Ravn, Location estimation for an autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry. *Proc. FUSION98 SPIE Conf.*, Las Vegas, Jul. 1998.

[4] H.-J. Von der Hardt, R. Husson, & D. Wolf, An automatic calibration method for a multisensor system: application to a mobile robot localization system. *Proc. IEEE International Conf. on Robotics and Automation, (ICRA'98)*, 1998, 3141–3146.

[5] A. Martinelli, N. Tomatis, A. Tapus, & R. Siegwart, Simultaneous localization and odometry calibration for mobile robot. *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS'03)*, Oct. 2003, 1499–1504.

[6] G. Antonelli, S. Chiaverini, & G. Fusco, A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation, *IEEE Trans. on Robotics 21*(5), 2005 994–1004.

[7] S. Murata and T. Hirose, Onboard locating system using real-time image processing for a self-navigating vehicle, *IEEE Trans. on Industrial Electronics, 40*(1), 1993, 145–154.

[8] K. Kato, H. Ishiguro, & M. Barth, Identifying and localizing robots in a multi-robot system environment. *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS'99)*, 1999, 966–971.

[9] J. Leonard and H. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, *IEEE Trans. on Robotics and Automation, 7*(3), 1991, 376–382.

[10] C.-C. Tsai, H.-H. Lin, & K.-H. Wong, Laser-based position recovery of a free-ranging automatic guided vehicle. *Proc. IEEE International Conf. on Mechatronics(ICM '05)*, July 2005, 1–6.

[11] H.-H. Lin, C.-C. Tsai, & J.-C. Hsu, Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering, *IEEE Trans. on Instrumentation and Measurement, 57*(9), 2008, 2024–2034.

[12] I. Rekleitis, G. Dudek, & E. Milios, Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. *Proc. International Joint Conf. in Artificial Intelligence (IJCAI)*, 1997.

[13] S. Roumeliotis & G. Bekey, Distributed multirobot localization, *IEEE Trans. on Robotics and Automation, 18*(5), 2002, 781–795.

[14] S. Thrun, Simultaneous localization and mapping. *Robotics and cognitive approaches to spatial mapping* (Berlin / Heidelberg: Springer, 2008), 13–41.

[15] F. Lu & E. Milios, Robot pose estimation in unknown environments by matching 2d range scans, *Journal of Intelligent and Robotic Systems, 18*(3), 1997, 249–275.

[16] C. Stachniss, Exploration and mapping with mobile robots. Ph.D. dissertation, University of Freiburg, Department of Computer Science, April 2006.

[17] S. Se, D. Lowe, & J. Little, Vision-based global localization and mapping for mobile robots, *IEEE Trans. on Robotics, 21*(3), 2005, 364–375.

[18] R. E. Kalman, A new approach to linear filtering and prediction problems, *ASME Journal of Basic Engineering, 82*, 1960, 35–45.

[19] M. Athans, R. Wishner, & A. Bertolini, Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements, *IEEE Trans. on Automatic Control, 13*(5), 1968, 504–514.

[20] D. Jung, J. Heinzmann, & A. Zelinksy, Range and pose estimation for visual servoing of a mobile robot. *Proc. IEEE International Conf. on Robotics and Automation (ICRA'98)*, May 1998, 1226–1231.

[21] S. Hoyt, S. Mckennoch, & L. G. Bushnell, An autonomous multi-agent testbed using infrared wireless communication and localization. Technical Report, Dept of EE, University of Washington, UWEETR-2005-0005, 2005.

[22] T. Chung, V. Gupta, J. Burdick, & R. Murray, On a decentralized active sensing strategy using mobile sensor platforms in a network. *Proc. IEEE Conf. on Decision and Control (CDC'04)*, Dec. 2004, 1914–1919.

[23] L. Slafer, The use of real-time, hardware-in-the-loop simulation in the design and development of the new Hughes HS601 spacecraft attitude control system. AIAA Technical Library, NTIS HC A20/MF, 1989.

[24] W. Jin & X. Zhan, A modified Kalman filtering via fuzzy logic system for ARVs location. *Proc. International Conf. on Mechatronics and Automation (ICMA'07)*, Aug. 2007, 711–716.

[25] G. K. Fricke, D. Milutinović, & D. P. Garg, Sensing and estimation on an experimental testbed for swarm robotics. *Proc. ASME Dynamic Systems and Controls Conf. (DSCC'09)*, to appear Oct. 2009.

[26] V. Braitenberg, *Vehicles: Experiments in synthetic psychology*. (Cambridge, Massachusetts: MIT Press, 1984).

A.2.6 Discrimination and Tracking of Individual Agents in a Swarm of Robots

The following paper (on the next 6 pages) was co-authored by Gregory K. Fricke and Devendra P. Garg, and appeared in the Proceedings of the 2010 American Control Conference on pages 2742–2747, published and presented in July 2010.

2010 American Control Conference
Marriott Waterfront, Baltimore, MD, USA
June 30-July 02, 2010

Discrimination and Tracking of Individual Agents in a Swarm of Robots

Gregory K. Fricke, Devendra P. Garg
Department of Mechanical Engineering
and Materials Science
Duke University
Durham, NC 27708
Email: {gregory.fricke, dpgarg}@duke.edu

Abstract— **Multi-agent robotic systems utilizing one-to-one communication between individuals necessarily require that each robot has a unique identity, such that shared information is appropriately utilized. When used with a central computer and external vision system, various methods may be used to identify individuals, including unique patterns. This requires *a priori* knowledge of the patterns, while taxing the vision system. Such methods are discussed along with their limitations. An alternative method is proposed to alleviate the strain on the vision system, eliminating the need for prior knowledge of individual patterns.**

I. INTRODUCTION

In multiple-robot systems utilizing one-to-one communication, robots must be assigned unique identities within the system to facilitate appropriate message transmission. In many cases, robots using wireless communication will have a unique identity established via some hardware method, such as a multi-bit switch, or a MAC address.

For a central supervisor to provide feedback and unique information to specific members of a swarm, the system must be able to identify the robots, associating a known or observed client ID with a detected and tracked agent. Such implementations may be used in applications such as soccer-playing, area-surveillance or -defense, with the central supervisor or controller managing such tasks as mission planning, mode-switching, map construction, or message switchboarding. For information exchange, the identity of both sender and receiver must be known.

The identification process may be addressed through the use of special tracking tools. High-cost vision systems utilizing custom fiducial markers on each robot are available, reliable, and capable of tracking both air- and ground-based robots (see, e.g., [1]). It is the goal of this research work to find a method requiring little or no specialized equipment.

A method is presented that relies on data reduction to provide information for discrimination. The method thus reduces to an estimation not unlike a least-squares fit. A complete presentation of a system utilizing such a method is given, including the dependencies between estimation components.

II. IDENTIFICATION VIA VISUALLY DISCRIMINATORY INFORMATION

In collective-robotic applications with a central observer or controller, it is common to use features on the robots to provide identity information via features detectable by a visual sensor. Some examples include color/shape patterns on RoboCup "players" [2] or other swarm robots [3], 2D bit-patterns on unmanned ground vehicles (UGVs) in a swarm implementation [4], and many others.

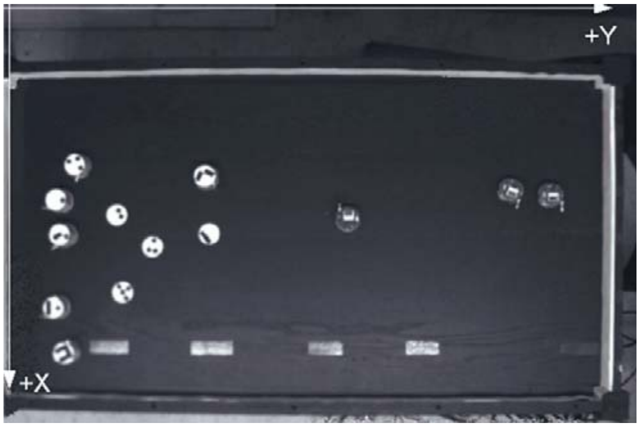


Fig. 1. Overhead view of Khepera-II arena in the RAMA Lab.

The Robotics and Manufacturing Automation (RAMA) Laboratory at Duke University has developed a modular testbed for conducting multi-robot experiments [5], using Khepera-II robots, a Cognex Insight 5400 real-time industrial machine vision system, and two SICK LMS400-1000 LADAR sensors.

The SICK LADAR sensors provide only range data, and thus the agents must be identified as features in the data. An arc-detection method developed at the RAMA Lab locates agents within a set of range data [5]. The measurements of the extracted agents include only x-y position data, as the Khepera-II robots are cylindrical with no distinguishing features to be used to measure orientation.

The Cognex system is comprised of the camera, which is a standalone computer performing image capture and processing, and a desktop computer for control of the camera. With no image processing, The camera provides black and white images live video at up to $40 fps$ with negligible lag. Pixel resolution ranges from $2.43 mm/pix$ to $2.75 mm/pix$ across the arena.The camera may be configured to perform certain image-processing tasks such as identification of user-defined patterns. Fig. 1 gives a sample image of the arena.

978-1-4244-7427-1/10/$26.00 ©2010 AACC

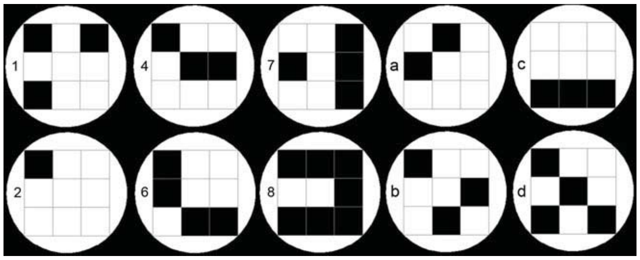


Fig. 2. Unique IDs for Cognex identification via pattern-recognition

1) Nine-bit Identifier: In order to generate position and orientation data along with unique identification using this camera system, distinguishing features were added to the robots. Several 9-bit patterns, generated on $2in$ diameter circles, are shown in Fig. 2. This feature size, exceeding $5pix$, was thought to be sufficient to guarantee robustness in pattern matching.

Baseline tests were performed with ten patterns to select a few patterns that would work together. In these tests, all of the pattern discs were placed in the arena, closely spaced. The pattern-recognition software was then enabled for one pattern at a time. This series of tests was performed with the patterns grouped in two regions of the arena: towards the left-most edge (min-y) of the arena, and clustered around the middle of the arena.

For a swarm of only four members, using patterns 4, 6, 7, and 8, the data sample rate was reduced to approximately 1 frame/sec with nearly 2 seconds of lag. The use of this type of pattern recognition for identification and orientation clearly places a severe limitation on the scalability of the swarm.

Difficulty in recognition arises via blur due to focus that, as evident in Fig. 1, varies throughout the field of view (FOV), and off-orthogonal pixelation that occurs when the robot is askew to the cardinal axes of the camera system. In these cases, the sharply drawn grid boundary lines are no longer well-defined. This coupled with the non-linear pixel scaling towards the edges of the arena easily spoofs the pattern-recognition software. Given these limitations, an alternative method for identification is sought.

The burden on the vision system is reduced by utilizing a simple background contrast comparison and area (i.e., blob size) and shape (i.e., circular) parameters. In order to provide ample contrast against the black background of the arena, identical blank discs of paper are attached to the uppermost turret on each robot. This negates the measurement of orientation; the only measurement returned by the vision system is position data. This additionally means that the multiple agents present in the field of view are no longer distinguishable. This utilization of the vision system allows significantly faster image sample rate, achieving 10Hz without significant lag.

The task of identification becomes an estimation problem. The identification of individuals will be accomplished via comparison of the central supervisor's state estimates with the individual agents' dead-reckoning estimates. The agents have limited memory, so the shared data must be limited, i.e., it should not contain a full time history of estimates. For this estimation to work, several independent but complementary estimation methods must be applied.

III. Acquisition, Localization, and Tracking for Identification

The solution proposed involves the interaction of several seemingly independent tasks. The external observer must acquire the swarm members in the image as blobs. External localization is in some cases as simple as mapping a measurement to an observed robot; in this case, however, we seek to estimate the unobservable orientation and predict future motion. Tracking, in our case, refers to the correct association of measurements when multiple robots are present. The agents must maintain estimates of their own positions utilizing dead-reckoning. These components will be applied concurrently, resulting in unique associations of communication IDs with external location estimates.

A. External Localization

The central supervisor uses an Extended Kalman Filter for observation and estimation of four states for each agent. Once the set of individuals is defined, the central supervisor uses the measurements as updates to the estimated state. Individual agents in this swarm are characterized by the state vector $X = [x, y, v, \theta]^T$, where x and y are the typical coordinates in the arena reference frame, v is the linear speed, and θ is the orientation or heading where $\theta = 0$ corresponds to the x-axis.

Direct measurement of the first two states is possible given the sensors available in this experiment. The sensors are imperfect and the measurements include noise, so an appropriate measurement-blending filter should improve results. The third state, velocity v, cannot be directly measured with this set of position sensors but could be discretely estimated given a sufficiently short sample time. The final state, θ, cannot in any way be directly measured, requiring a state-blending filter to exploit the model (in particular, the non-holonomic constraints) allowing θ to be estimated.

To preserve scalability of the swarm by limiting required high update rate agent-to-host communication, the control input to the individual robots is unknown to the external observer. Bar-Shalom [6] offers multiple methods for dealing with such *maneuvering targets*: modeling the maneuver input as a random white noise process or autocorrelated Markov noise; estimating the input to correct the state estimate; or augmenting the state with the maneuver input to be estimated.

To propagate the model, changes in states v and θ are attributed to process noise, assumed to be zero-mean Gaussian-distributed white-noise with covariance ξ_v and ξ_θ respectively. These process noise parameters thus account for the unmodeled and unmeasured control signal. By including these contributions as noise, the magnitude of the parameters are larger than required, but with the benefit of a method that requires no communication with the robot. The state transition and measurement models are defined in Eqs. (1)-(3).

$$\mathbf{F}(X(t)) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

$$\dot{X}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{bmatrix} = \mathbf{F}(X(t)) + \begin{bmatrix} 0 \\ 0 \\ \xi_v(t) \\ \xi_\theta(t) \end{bmatrix} \tag{2}$$

The measurements provided by the vision system or LADAR pair are absolute position measurements, assumed to include zero-mean Gaussian-distributed white noise w_x, w_y with covariance W_x, W_y. Measurements are denoted x_m, y_m.

$$Z = \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \end{bmatrix} \tag{3}$$

These estimates are propagated and updated utilizing a 2nd-order Kalman Filter [7], [8]. The Jacobian A of the state transition model is given in Eq. (4). The Hessian matrices, denoted as F_1, F_2, F_3, F_4, of each state's transition function are shown in Eq. (5), where subscript k indicates the iteration step. Note that, as the third and fourth rows (f_3 and f_4) of the state-transition function depend only on the noise parameters and not any of the states, $F_3 = F_4 \equiv 0$.

$$A_k = \begin{bmatrix} 1 & 0 & \Delta t\cos(\theta_k) & -v_k\Delta t\sin(\theta_k) \\ 0 & 1 & \Delta t\sin(\theta_k) & v_k\Delta t\cos(\theta_k) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\begin{aligned} F_{1,k} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\Delta t\sin(\theta_k) \\ 0 & 0 & -\Delta t\sin(\theta_k) & -v_k\Delta t\cos(\theta_k) \end{bmatrix} \\ F_{2,k} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta t\cos(\theta_k) \\ 0 & 0 & \Delta t\cos(\theta_k) & -v_k\Delta t\sin(\theta_k) \end{bmatrix} \end{aligned} \tag{5}$$

The *prediction* step is given in Eqs. (6) and (7), which is used at each iteration to update states before applying the correction. These estimates are propagated and updated in typical form. The propagation and update equations are presented without derivation; additional details may be found in references [7] and [8].

$$\begin{aligned} \bar{X}_{k+1} = \hat{X}_k &+ \mathbf{F}(\hat{X}(t))\Delta t \\ &+ \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T tr(F_1 P_k) \\ &+ \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T tr(F_2 P_k) \end{aligned} \tag{6}$$

$$\bar{P}_{k+1} = A_k P_k A_k^T + Q_k \tag{7}$$

$$Q_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \Xi^v & 0 \\ 0 & 0 & 0 & \Xi_k^\theta(v) \end{bmatrix} \tag{8}$$

Without measurements, the system propagates via the state transition nonlinear function. In update steps where measurements are available and valid, the *innovation* is performed as shown in Eq. (9).

$$\begin{aligned} S_{k+1} &= C\bar{P}_{k+1}C^T + R \\ K_{k+1} &= \bar{P}_{k+1}C^T S_{k+1}^{-1} \\ \hat{X}_{k+1} &= \bar{X}_{k+1} + K_{k+1}(Z_{k+1} - C\bar{X}_{k+1}) \\ P_{k+1} &= (I - K_{k+1}C)\bar{P}_k \end{aligned} \tag{9}$$

$$R = \begin{bmatrix} W_x & 0 \\ 0 & W_y \end{bmatrix} \tag{10}$$

In Eq. (7) and Eq. (8) above, the process noise covariance matrix (more specifically, Ξ_k^θ) is indicated as a function of time-step k. This is due to the observation that as $v(t) \rightarrow 0$, the uncertainty in $\hat{\theta}$ rises due to the dynamics of the robot. For this reason, Ξ_k^θ is defined as a function of v, such that it achieves its maximum value for $v = 0$ and decreases monotonically to $\Xi_k^\theta = 0$ at $v = \bar{v}$. For more details of this state-varying-noise adaptation, along with experimental results, see [9].

B. Tracking

Track adherence is critical to maintaining good estimation by ensuring that measurements are applied properly to the estimates of the correct agents. In the presence of multiple, physically homogeneous agents in the field-of-regard for the LADAR and vision systems, the order of measurements may be rearranged from sample to sample. Additionally, there may be occasional excessive error in the measurements due to a sensor glitch or communication error, or false identifications returned either by the LADAR arc-detection or by the vision system.

Data validation in tracking has been studied for several years, in many cases alongside Kalman Filter methods and especially in related fields such as radar tracking. Reid's method of tracking multiple targets [10] provides clear applicability to this case, although this method does not account for *drop track*, the case where a target is no longer in the field of view. Bar-Shalom [6] covers many methods for single- or multiple- target tracking, including Reid's Multiple Hypothesis Tracking (MHT) algorithm. Cox and Hingorani [11] proposed a computationally efficient implementation of Reid's MHT algorithm, achieving optimal assignment results in $O(M^2)$ time.

A matching and validation algorithm has been developed to validate and reorder the data. The method draws inspiration from all of the above methods.

The data set is first passed through a "sudden jump" filter, which simply inspects that the data contains real values. The next step, target matching or data association, is based on the *Hungarian Algorithm* developed by Munkres [12]. [1]

Measurements exhibiting excessive error due to background clutter or sensor noise cannot be identified easily as described above. The method for final data validation is adapted from Leonard's localization method [13], based on the prior works of Reid [10] and Bar-Shalom [6]. At each prediction step k, a *validation gate* g is established around the estimated output of the i-th agent, $\hat{Z}_k^i = C\hat{X}_k$. The range of the validation gate is dependent on the estimated covariance of error in the prediction, S. The predicted state for the i-th agent $\hat{Z}_k^i$is then compared to the j-th measurement, Z_k^j.

$$V_{ij} = Z_k^j - \hat{Z}_k^i \tag{11}$$

$$g^2 \geq V_{ij} S_i^{-1} V_{ij}^T \tag{12}$$

The parameter g defines the region of acceptance; the validation region is χ^2-distributed and depends on the dimension of the measurement vector. For a 2-dimensional measurement vector, $g = \sqrt{4.61} \approx 2.15$.

C. Individual Localization

Individual self-localization requires the robot to have the ability to make appropriate measurements of its environment. In the scenario where *a priori* information about the environment is known and where the robot has the ability to make measurements with sufficiently high resolution, the self-localization problem has been extensively covered [13]–[17]. Simple robots, such as the Khepera-II, do not have such sensing capability and must rely on external position updates. To maintain the on-board position estimate, odometry is utilized. The odometry model should be calibrated in order to minimize the required rate of the external updates. Odometry calibration for this system is based on Antonelli's Least Squares Method [18], [19]. Implementation of this method is discussed in [5], [20].

Odometric error increases quadratically with distance [21], hence it is quite susceptible to model errors. Many methods for calibrating odometry or mitigating the associated errors have been proposed [15], [18], [19], [22], [23]. The sources of odometric error can be reduced to two subgroups: systematic error and non-systematic error. For any of these methods to work properly there must be a known position reference to allow correction. The systematic error can be reduced by appropriate parameterization of the odometry model, followed by application of an appropriate estimation method. With appropriate feedback, the non-systematic error can also be reduced [15].

For a differential-drive, two-wheeled robot, the approximate odometry equations may be written in discrete time as shown in Eq. (13) and Eq. (14). The position and orientation (or *pose*) of the robot are given by (x, y, θ) in world coordinates. For simplicity of frame mapping, the position of the robot is taken at the midpoint of the wheel axis, the natural center of rotation. The frame is right-handed, with θ measured positive about the z-axis. For incremental parameters, the subscript k denotes the change in value over the interval $[k-1,\ k]$. For cumulative parameters, the subscript k denotes the value at the end of the interval. The incremental angular rotation of each driveshaft is $\Delta\psi_k^{R/L}$. The wheel radii are represented as $r_{R/L}$. The encoder scale factor mapping encoder counts to radians of wheel shaft rotation is $c_{R/L}$. Incremental encoder measurements for each wheel are given by $\delta e_k^{R/L}$. The incremental distance traveled by each wheel is given by $d_k^{R/L}$. The separation of the points of contact of the wheels is given by b. The incremental change in orientation of the robot is given by $\Delta\theta_k$, and the incremental, scalar change in position is denoted ΔS_k.

$$\begin{aligned}
\Delta\psi_k^{R/L} &= \delta e_k^{R/L} c_{R/L} \\
d_k^{R/L} &= r_{R/L} \Delta\psi_k^{R/L} \\
\Delta\theta_k &= \frac{d_k^R - d_k^L}{b} \\
\Delta S_k &= \frac{d_k^R + d_k^L}{2}
\end{aligned} \tag{13}$$

$$\begin{aligned}
x_{k+1} &= x_k + \Delta S_k \text{sinc}(\frac{\Delta\theta_k}{2}) \cos(\theta_k + \frac{\Delta\theta_k}{2}) \\
y_{k+1} &= y_k + \Delta S_k \text{sinc}(\frac{\Delta\theta_k}{2}) \sin(\theta_k + \frac{\Delta\theta_k}{2}) \\
\theta_{k+1} &= \theta_k + \Delta\theta_k
\end{aligned} \tag{14}$$

where $\text{sinc}(x) \triangleq \frac{\sin x}{x}$, $\lim_{x \to 0} \text{sinc}(x) = 1$.

Additional navigation values of interest are the curvilinear abscissa S, the total direction-ignorant accumulated distance traveled S^{abs} ($S^{abs} \geq 0$), and the total orientation change from initial position θ_0. These are accumulated as shown in Eq. (15). The recording of these additional values allows the use of methods such as Antonelli's Least-Squares [18], [19].

$$\begin{aligned}
S_{k+1}^{abs} &= S_k^{abs} + |\Delta S_k| \\
S_{k+1} &= S_k + \Delta S_k \\
\theta_{k+1} &= \theta_k + \Delta\theta_k
\end{aligned} \tag{15}$$

Antonelli's method of odometry calibration [18], [19] starts with the mapping of measured driveshaft angular velocities $[\omega_R, \omega_L]^T$ to state parameters $[v, \dot{\theta}]^T$ (linear and angular speed) via the state-transition matrix C as shown in Eq. (16).

$$\begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r_R/2 & r_L/2 \\ r_R/b & -r_L/b \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = C \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \tag{16}$$

The method requires construction of two regressor from P trajectories, one for orientation and the other for position. This data is compared to start and end values of accumulated wheel encoder values. This offline odometry calibration method is adapted to perform in an online setting, with the individual robot following a trajectory consistent with its task (mapping, foraging, etc.). These trajectory segments

[1] Our implementation uses utilizes a software package written by Markus Buehren, updated 22 Oct 2008: http://www.mathworks.com/matlabcentral/fileexchange/6543

together are the P trajectories. With each new measurement, the regressor matrices are augmented, and a new $\hat{C}$ found. Implementation of such an algorithm on an on-board processor may be limited by memory restrictions (as the regressor matrices and observation vectors must be augmented at each time step) and processor capability (foremost, the ability to efficiently perform matrix inversion; at each update this inversion would involve a larger matrix).

A system using this method is constrained by the requirement of a fast external position-tracking system to accurately observe S; the position-tracking system's sampling rate defines the upper bound of the robots' velocities. The measurement system for this type of calibration must be capable of accurately measuring orientation, requiring good performance of the External Localization described above.

IV. Identification

The final task is to unify all of the above estimation methods to yield communication with multiple robots with whom communication is possible, yet whose positions are unknown either to themselves or the central observer.

It is desired to find a set of parameters that, under non-identical motion of agents, will provide discriminatory information. The proposed set of data to be used for identification is $[\hat{X}_k, \hat{S}(k)]$, the current state estimate and the current estimated magnitude of the estimated abscissa. The agents, however, do not know the true values of X_0 in the absolute reference frame, so this vector is actually the agent's estimate of the *changes* in state and abscissa estimate since sample $k = 0$.

This places an additional requirement on the external observer to maintain the cumulative navigation parameter $\hat{S}(k)$, as well as the initial positions of each robot.

A probability-based identification hypothesis for matching external estimates to on-board estimates of these proposed parameters can then be constructed. A simple implementation would be to once again utilize the matching and gate-validation method described above in Section III-B and in Eqs. (11)-(12). The cost inputs would be related to the differences between the external and internal estimates of $\hat{S}(k)$ and of $\Delta\hat{X}(k)$.

For sufficient confidence in the discrimination, the trajectories of the robots must necessarily be distinct. The external observer may be able to estimate a confidence factor by constructing the trajectory-richness parameters suggested by Antonelli [19] and discussed in Section III-C. As the condition numbers of these matrices grow, so should the confidence that the discrimination is correct.

All of the above estimation methods must perform reliably for this Identification to work. The External Localization and Tracking methods are tied closely together, as without one the other will not (and cannot) work. Without Tracking, the external observer cannot accurately accumulate the navigation parameters of individual robots. Communication is necessary to relate the internal estimates to the external observer. These information packets are small (five floating-point numbers), and the rate of communication required is very low. There would be little or no benefit to transmit these parameters to the central observer at high rate, as the system requires time to develop sufficient richness and the dynamics of the robots are quite slow.

V. Preliminary Experimental Results

For verification of this method, a more capable experimental setup is employed. The arena for robot exploration is an area that is roughly $5m$ x $3.5m$. The agents in this experiment are Create mobile robots from iRobot. The motion capture video system is comprised of twelve OptiTrack V100R2 cameras from NaturalPoint, and the tracking software utilizes the NaturalPoint Tracking Tools API. Each Create robot is assigned a unique pattern of reflective markers and a corresponding "rigid body" definition in the software. This allows fairly robust tracking of the agents at 100Hz.

The information made available to the supervisor is the same limited information described in the above sections, with the additional information available for use in verification and scoring of the method.

A sample set of trajectories is shown in Fig. 3. This data was gathered with five agents moving simultaneously. The control consisted of two components: a random (uniformly distributed) input for each linear acceleration (small, positive mean) and angular acceleration (zero-mean); and agent avoidance via simulated sensors. Velocity is capped at $V_{max} = 100mm/s$. Additionally, the agents are equipped with bump sensors; if activated, the agent reversed at maximum velocity, then rotated approximately 90° with the direction of rotation chosen arbitrarily.

Initial tests showed that parameter S is not a very good estimator unless the supervisor can very accurately determine orientation. In contrast, $\hat{X}$ and S^{abs} are quite simple to maintain by the supervisor, but with the control system used thus far does not provide a good differentiator between agents. This is due to the maximum velocity with small, positive mean acceleration. As a result, the agents spend most of their time at maximum velocity; hence, if all robots begin moving at the same time, the total accumulated distance traveled at time T for each robot approaches $T * Vmax$.

Additional limitations in testing have arisen due to communication limitations in the contol of each individual robot. In this implementation, a single computer performs all of the work. Five identical instances of the robot control code execute simultaneously, each one communicating with a different robot via Bluetooth. Under this heavy traffic load, communication between the robot hardware and its "virtual controller" is inconsistent, and the data is occasionally corrupted. This problem is being addressed via the introduction of on-board computers on each robot. Gumstix Verdex computers will be interfaced with each robot via a Sticky Interface from ElementDirect, allowing the robots to truly be independently controlled. Communication with the supervisor can then occur via the more reliable wireless Ethernet interface.

Additional simulations utilizing Player/Stage [24] are under way to confirm the success of this method and to assess

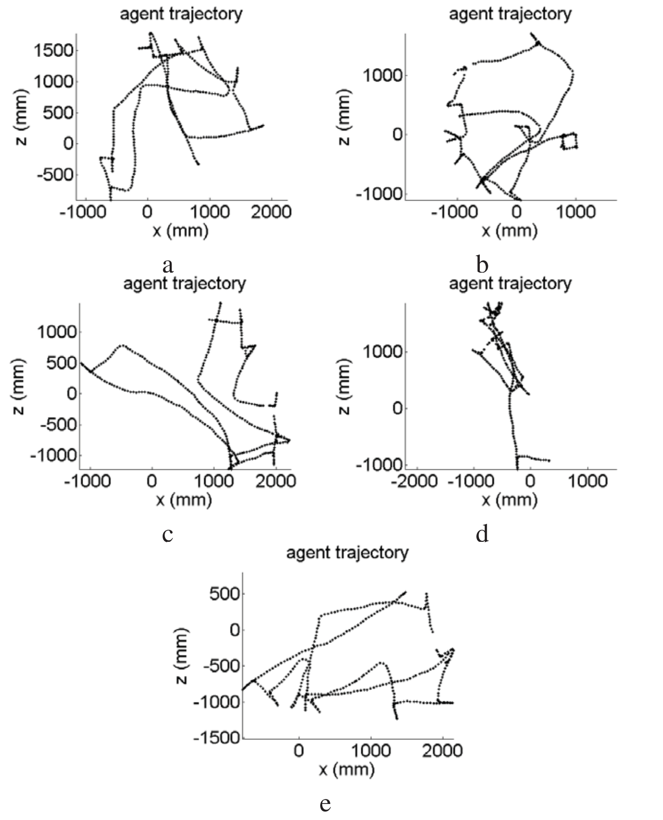


Fig. 3. Characteristic trajectories for a set of five agents moving randomly in the arena.

processing limitations with greater numbers of robots. The simulations also allow a greater range of testing parameters.

VI. CONCLUSION

A method for inferential identification has been proposed based on prior experiments and on-going simulation. This identification allows the establishment of external feedback for a collection of visually indistinguishable robots, which in turn allows for odometry correction and calibration, and orientation estimation. These methods require no special equipment or exceptional machine vision processing. The results are also marginally scalable, with limitations only in the matching phase (the Hungarian Method) of the tracking algorithm.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Army Research Office under grant number W911NF-08-1-0106.

REFERENCES

[1] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, “Real-time indoor autonomous vehicle test environment,” *Control Systems Magazine, IEEE*, vol. 28, no. 2, pp. 51–64, April 2008.

[2] M. Simon, S. Behnke, and R. Rojas, *RoboCup2000: Robot Soccer World Cup IV*. Springer Berlin / Heidelberg, 2001, ch. “Robust Real Time Color Tracking”, pp. 239–248.

[3] S. Hoyt, S. Mckennoch, and L. G. Bushnell, “An autonomous multi-agent testbed using infrared wireless communication and localization,” Dept of EE, University of Washington, Tech. Rep. UWEETR-2005-0005, 2005.

[4] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, “Decentralized cooperative control - a multivehicle platform for research in networked embedded systems,” *Control Systems Magazine, IEEE*, vol. 27, no. 3, pp. 58–78, June 2007.

[5] G. K. Fricke, D. Milutinović, and D. P. Garg, “Sensing and estimation on an experimental testbed for swarm robotics,” in *Proc. ASME Dynamic Systems and Controls Conference (DSCC’09)*, Oct. 2009.

[6] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional, Inc., 1988.

[7] M. Athans, R. Wishner, and A. Bertolini, “Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements,” *Automatic Control, IEEE Transactions on*, vol. 13, no. 5, pp. 504–514, Oct. 1968.

[8] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering, ASME*, vol. 82, pp. 35–45, 1960.

[9] G. K. Fricke, D. Milutinović, and D. P. Garg, “Robotic pose estimation via an adaptive Kalman Filter using state-varying noise,” in *Proc. IASTED Robotics and Automation Conference (IASTED RA’09)*, Cambridge, MA, Nov 2009.

[10] D. Reid, “An algorithm for tracking multiple targets,” *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, Dec 1979.

[11] I. Cox and S. Hingorani, “An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 2, pp. 138–150, Feb. 1996.

[12] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[13] J. Leonard and H. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[14] K. Kato, H. Ishiguro, and M. Barth, “Identifying and localizing robots in a multi-robot system environment,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’99)*, vol. 2, 1999, pp. 966–971.

[15] A. Martinelli, N. Tomatis, A. Tapus, and R. Siegwart, “Simultaneous localization and odometry calibration for mobile robot,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’03)*, vol. 2, Oct. 2003, pp. 1499–1504.

[16] S. Roumeliotis and G. Bekey, “Distributed multirobot localization,” *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[17] ——, “Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’00)*, vol. 3, 2000, pp. 2958–2965.

[18] G. Antonelli, S. Chiaverini, and G. Fusco, “A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation,” *Robotics, IEEE Transactions on*, vol. 21, no. 5, pp. 994–1004, Oct. 2005.

[19] G. Antonelli and S. Chiaverini, “Linear estimation of the odometric parameters for differential-drive mobile robots,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IRAS’06)*, Oct. 2006, pp. 3287–3292.

[20] G. K. Fricke, “Localization, tracking, and odometry calibration of a multi-agent swarm system,” Master’s thesis, Duke University, Department of Mechanical Engineering and Materials Science, Durham, NC, April 2009.

[21] A. Kelly, “General solution for linearized systematic error propagation in vehicle odometry,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’01)*, vol. 4, 2001, pp. 1938–1945.

[22] K. S. Chong and L. Kleeman, “Accurate odometry and error modelling for a mobile robot,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’97)*, vol. 4, Apr 1997, pp. 2783–2788.

[23] T. D. Larsen, M. Bak, N. A. Andersen, and O. Ravn, “Location estimation for an autonomously guided vehicle using an augmented Kalman Filter to autocalibrate the odometry,” *FUSION98 SPIE Conference, Las Vegas*, Jul. 1998.

[24] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The Player/Stage project: Tools for multi-robot and distributed sensor systems,” in *Proc. 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.

A.2.7 Mobile Sensor Frame Mapping via Vision and Laser Scan Matching

The following paper (on the next 4 pages) was co-authored by Gregory K. Fricke, Adam W. Caccavale, and Devendra P. Garg, and appeared in the Proceedings of the 2010 International Symposium on Resilient Control Systems on pages 43–46, published and presented in August 2010.

Mobile Sensor Frame Mapping via Vision and Laser Scan Matching

Gregory K. Fricke
Adam W. Caccavale
and Devendra P. Garg
Department of Mechanical Engineering and Materials Science
Pratt School of Engineering, Duke University
Durham, NC 27708
Email: {gregory.fricke, adam.caccavale, dpgarg}@duke.edu

***Abstract*—Cooperative mobile robots must have knowledge of their positions relative to the group in which they are operating. Common on-board sensors such as laser rangefinders may be used to detect and track other robots with high precision, though limited feature recognition and susceptibility to occlusion reduces the efficacy of this solution alone. Matching multiple robots' laser scans can overcome some of these issues, but requires extensive memory usage and large communication bandwidth. Overhead imaging systems may also be utilized, though sensor nonlinearities, field of view restrictions, and data latency limit such usage.**

A data fusion method is proposed for dynamically evaluating a mobile robot's position by matching laser scan data to overhead image data.

I. Introduction

For autonomous mobile robots, whether acting independently or within a swarm, knowledge of its position within the environment is critical for successful navigation and conduct of tasks. This position may be relative to features in the environment, such as the starting position, goal, or other arbitrary point, or may be relative to other members of the swarm. If a supervisory positioning system is available, localization may become a trivial task, although the speed and reliability of communication arises as a limiting factor. To mitigate the necessity of high-speed communication as well as to allow scalability of a swarm (where the communication and perceptive requirements of the central supervisory member quickly exceed capability), the individual mobile member should be tasked with additional localization responsibility.

For the specific case of formation control, the robot's position relative to the formation is as important, if not more so, than the position relative to the workspace. The works of Yamaguchi and Burdick [1], Desai *et al* [2], Tanner *et al*, [3]–[6], and Antonelli [7] provide multiple methods for controlling formations of robots under various sensing and communication scenarios.

Published research in collective or collaborative localization includes that of Fenwick, Newmann, and Leonard, [8], Fierro, *et al* [9], Howard, Matarić, and Sukhatme [10], Rekleitis, Dudek, and Milios, [11], and Roumeliotis and Bekey [12], [13]. A thorough discussion of the necessity of localization is provided by Negenborn in his Master's Thesis [14].

Laser rangefinders have become ubiquitous on-board sensors for mobile robots in the last decade due to their high accuracy (for most common models, the error is less than $1cm$ over the usable range), large field of view (frequently $> 180\,^{\circ}$), and high speed data collection ($10Hz$ or faster for full scan). Laser rangefinders seem to be the sensors of choice for simultaneous localization and mapping applications [15].

Laser rangefinders have two major limitations in multi-robot applications, however. The most severe is the problem of occlusion. Since these sensors make measurements in the same plane as the robot (or a parallel plane), only the nearest object to the rangefinder at that planar height can be measured. This is illustrated in Fig. 1. The secondary problem is the sparsity of information contained in the data. That is, the rangefinder measures the nearest return surface, but categorizing that surface requires specialized processing.

Overhead imaging systems may seem to be an end-all solution to the problem of formation measurement and off-board localization. In fact, such systems do in many cases simplify measurements of robot locations. Machine vision software, such as OpenCV [16], [17], is readily available for basic feature extraction and pattern matching. These systems, however, also have limitations in processing time, calibration requirements, and *a priori* knowledge requirements such as the definitions of patterns.

II. Problem Description

A method is sought to localize all mobile members of the collective relative to a common frame. In this work, we provide an architecture for relative localization of members of a robot collection by the fusion of various sensor data sets. This method is based on a combination of multiple laser rangefinders making in-plane measurements of the environment (which includes neighboring robots) with one or more optical overhead images taken by a supervisory observer.

Foremost is the problem of data processing. In order to provide accurate measurement capability, digital images must be of high pixel resolution, placing stiff requirements for data bandwidth, processing power, and memory space. Memory requirements increase with the square of pixel resolution (for fixed field-of-view), and processing power requirements grow

978-1-4244-5954-4/10/$26.00 ©2010 IEEE

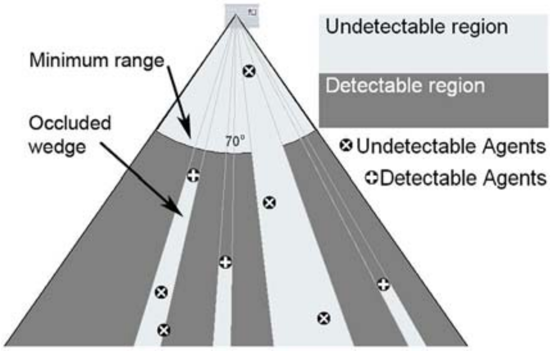


Fig. 1. The occlusion problem for laser rangefinders. The scale is based on a SICK LMS400 detecting Khepera-II robots.

at least as fast. Sensors capable of high-resolution, high-bandwidth imaging such as this can be quite expensive, as is the processing required to turn these images into usable data for the robots at sufficient data rates.

Secondary and tertiary issues with using overhead imaging are the calibration of the imaging system and the finite field of view of the system. We seek the implementation of an overhead imaging system mounted to an aerial robot, meaning that the imaging sensor frame is explicitly *not* fixed relative to the robotic workspace. Imaging sensors may incorporate zoom lenses which dynamically change the effective pixel resolution; with the exception of very specialized cameras, however, the field of view is generally limited to around $50\,^\circ$. In order to capture a desired planar area, the altitude at which the UAV-mounted sensor must operate is dictated. The planar resolution of the camera is a function of the distance to the plane, however, so a trade arises between maximized coverage area and optimal planar pixel resolution. Related to this is the common problem of nonlinear planar pixel scaling; for most digital imagers, there may be large differences between the planar pixel scale factor along the nadir or line-of-sight axis of the camera and the same scale factor at the edges of the field of view. This scale factor variation is exacerbated by larger angular fields of view, i.e., as the angle from the focal plane to the sensed plane increases.

A sample image taken from an overhead machine-vision sensor is given in Fig. 2. This image is cropped; the center of the true image field of view is to the right of center in this image. Note the apparent curvature of the arena wall on the left of the image, and the projection of the robot bodies at the lower extent of the image. These projection issues can induce estimation errors in the true locations of the robots within the field, and, if pattern matching is utilized for robot discrimination, may severely reduce the rate of success (see, e.g., [18]).

III. Proposed Method

The first phase of estimation is performed by each individual mobile sensor. Each robot i detects other robots within its laser scan field of regard. For this, the arc-detection method developed by the authors (described in [18], [19]) is employed.

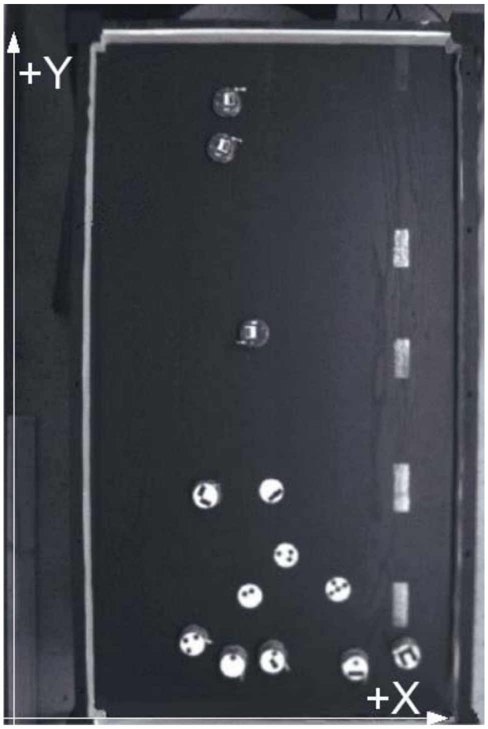


Fig. 2. Sample overhead image from Cognex 5400 camera of a rectangular arena with Khepera-II robots extending above a planar surface.

A geometric graph $\mathbf{S}_i$ is assembled from these relative measurements. This subgraph of the total robot collective graph may not be observable by any one sensor.

In similar fashion, the overhead observer performs simple blob-detection or pattern-matching to identify robots within its field-of-view, and constructs a geometric graph $\mathbf{G}$ of all detected robots.

The crux of the problem is then the data assembly to combine all of these subgraphs into the geometric graph representing the entire system. In this initial work, we will assume that the overhead observer can detect the entire collection of robots, implying that the complete graph is known, though it includes sensing error. We denote the complete graph $\mathbf{G}$ and the subgraph detected by agent i as $\mathbf{S}_i$.

The method currently employed is an exhaustive comparison of the subgraph in question with the complete graph. This problem, called *subgraph isomorphism*, is unfortunately among the list of proven NP-complete problems [20].

The algorithm is executed as following. First, the ranks of the graph and subgraph, $r_{\mathbf{G}}$ and $r_{\mathbf{S}_i}$, respectively, are established. The assumption that $\mathbf{S}_i \subset \mathbf{G}$ implies $r_{\mathbf{G}} \geq r_{\mathbf{S}_i}$, thus the set of possible nodes of $\mathbf{G}$ included in $\mathbf{S}_i$ is $\mathbf{I} = \{1, ..., r_{\mathbf{G}}\}$. We then generate a list of hypotheses, permutations of $r_{\mathbf{S}_i}$ indices chosen from set $\mathbf{I}$.

The algorithm then loops over the set of hypotheses, $\mathbf{I}$, testing each hypothesis. The test criterion is to identify the hypothesis, $\mathbf{I}_n \subset \mathbf{I}$, which minimizes the difference between the truncated graph $\mathbf{G}$ with the subgraph $\mathbf{S}_i$. That is,

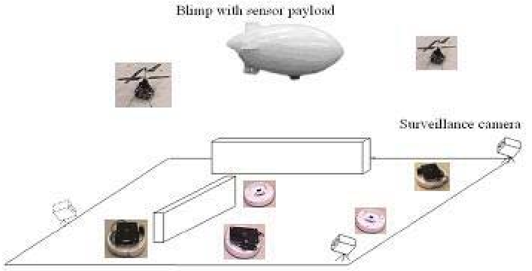


Fig. 3. Graphical depiction of the lab environment

$$error = ||\mathbf{G}_n - \mathbf{S}_i \times \frac{||\mathbf{G}_n||}{||\mathbf{S}_i||}|| \qquad (1)$$

where $\mathbf{G}_n$ is the submatrix of $\mathbf{G}$ corresponding to the index set $\mathbf{I}_n$.

IV. Experimental Setup

We are utilizing three different experimental configurations to verify this method. The first, described in detail in [18], utilizes a fixed overhead Cognex 5400 machine-vision sensor and computer observing a $1.6m \times 0.8m$ arena containing up to twelve Khepera-II robots. Additional sensing is provided by a pair of SICK LMS400 LADAR sensors. This system with fixed, external sensors is used to measure the accuracy and precision of this method, as this system is well-characterized.

The second experimental configuration, depicted in Fig. 3, is a larger lab space ($5.5m \times 5.3m$), utilizing several iRobot Create ground vehicles. The Creates are each equipped with an on-board computer (currently, Dell Mini10 netbooks), a webcam (Logitch QuickCam Pro 9000), and a laser rangefinder (Hokuyo URG-04LX-UG01). Additional Create robots are planned with similar sensing but utilizing GumStix Verdex on-board computers. Overhead imaging is provided by one or two Skybotix CoaX helicopters each equipped with a webcam (Targus AVC05EU) and an on-board computer (GumStix Overo). Additionally, our research group is developing a semiautonomous indoor blimp, equipped with an on-board computer (GumStix Verdex) and I/O board (Gumstix RoboStix) to support a webcam. For metrology as well as possible augmentation of the vision capabilities, this lab space is equipped with a 12-camera NaturalPoint OptiTrack motion tracking system. A Create robot in our experimental configuration is shown alongside a CoaX aerial robot in Fig. 4.

The third experimental configuration is a simulated world, implemented in Player/Stage [21]. The simulation allows us a path for software development with fewer uncertainties than are usually encountered with real hardware systems. It also provides us with the ability to predict whether this method will be successful and estimates of error bounds, as well as to evaluate the effects of different types of sensors to be procured in the future.

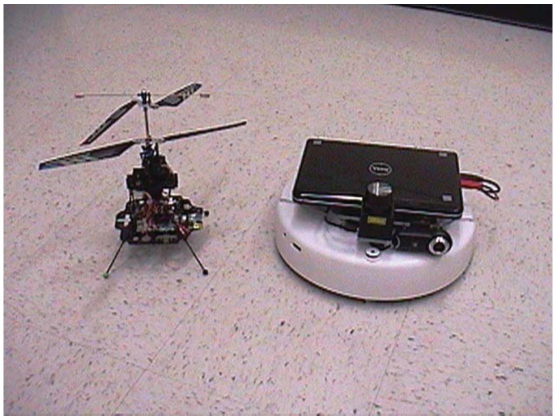


Fig. 4. The Skybotix CoaX helicopter with one iRobot Create in experimental configuration.

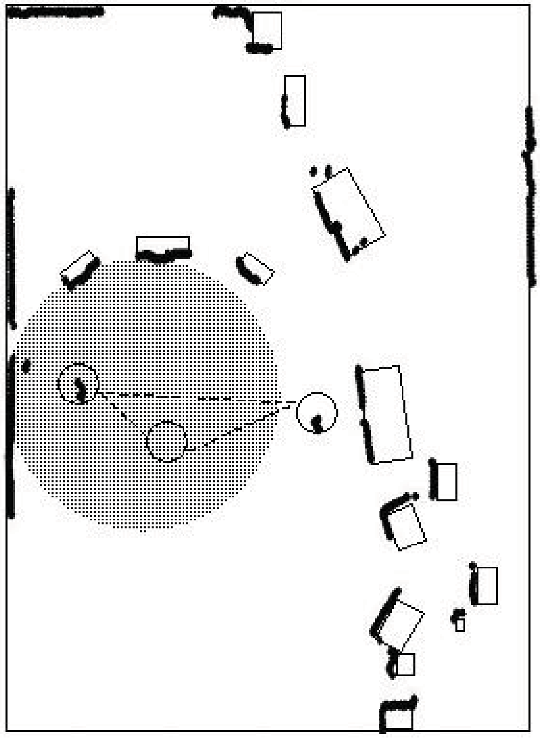


Fig. 5. On-board laser data superimposed onto sketchup of lab layout, including obstacles and other robots. The grey circle indicates a typical field of view for a camera field-of-view on one of the aerial vehicles.

The strengths of each sensor type should be exploited to improve the estimation quality of the system. The utilization of multiple in-plane laser rangefinders on-board individual robots extends the sensed area. Including overhead images fills gaps where the rangefinders are occluded. Use of the overhead image provides a template onto which the rangefinder data may be matched, yielding a common frame of reference for all vehicles in the collective.

The ultimate goal of this matching is to provide a method of unifying multiple sensor frames into a single relative frame, including correction for rotation, translation, and scaling. As of this writing, the problem of skewing or stretching has not yet been addressed.

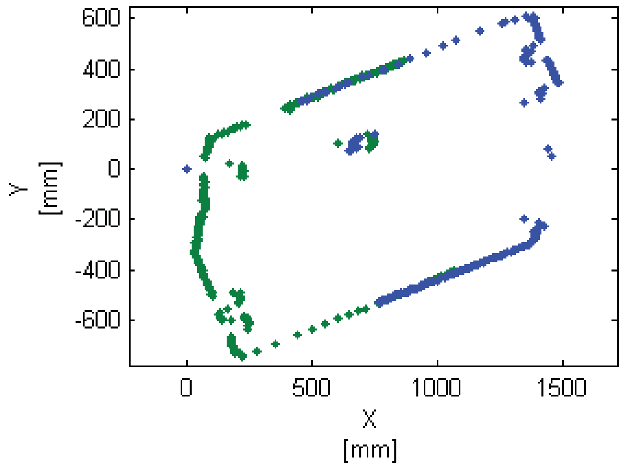


Fig. 6. Two laser rangefinder scans, correlated into a single map. The difficulty in identifying robots in this data is made clear; there are six robots in the field in these scans.

V. Results

The most obvious downfall of this method is the assumption that the graph topography does *not* contain repeated subgraph isomorphisms of rank $r_{\mathbf{S}_i}$. I.e., the elements of the graph must be sufficiently diverse. This assumption leads to the practical requirement that subgraph $\mathbf{S}_i$ is *large enough*. Exactly how large $r_{\mathbf{S}_i}$ must be depends on the diversity of the full graph $\mathbf{G}$. It should be clear that this method cannot work for very regular formations, such as a rigid formation of equilateral triangles.

If the above condition is met (i.e., $r_{\mathbf{S}_i}$ is large enough and $\mathbf{G}$ is diverse enough), and when graph $\mathbf{G}$ and subgraph $\mathbf{S}_i$ contain no error, any matrix norm works with this method. Indeed, even the scaling of $\mathbf{S}_i$ is unnecessary if the sensor frames being matched have uniform scaling. If noise is included, the Frobenius norm works most reliably, as it is a corollary to the sum-of-squared-error in vector space.

VI. Conclusion

This work is on-going with data collected in the Khepera-II lab, along with new experiments and data collection in the Create lab and in the Player/Stage simulation. Preliminary results are promising, though the parameter space has yet to be explored for robustness and processing efficiency – specifically the scalability of this problem. If this method is to be used with a collective or swarm robotic system, the processing requirements must be fixed (or nearly so) with increasing numbers of robots, requiring that each individual robot must do as much processing as possible. Continuing work focuses on more efficient methods, such as the search algorithms in [22] and the use of graph topology for image processing [23].

Acknowledgment

The authors gratefully acknowledge the support of the Army Research Office under grants W911NF-08-1-0106 and W911NF-09-0307.

References

[1] H. Yamaguchi and J. Burdick, “Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations,” *Proc. 1998 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3573–3580, May 1998.

[2] J. Desai, J. Ostrowski, and V. Kumar, “Controlling formations of multiple mobile robots,” in *Proc. 1998 IEEE International Conference on Robotics and Automation*, vol. 4, May 1998, pp. 2864–2869.

[3] H. Tanner, A. Jadbabaie, and G. Pappas, “Stable flocking of mobile agents, part I: fixed topology,” in *Proc. 42nd IEEE Conference on Decision and Control (CDC2003)*, vol. 2, Dec. 2003, pp. 2010–2015.

[4] ——, “Stable flocking of mobile agents part II: dynamic topology,” in *Proc. 42nd IEEE Conference on Decision and Control (CDC2003)*, vol. 2, Dec. 2003, pp. 2016–2021.

[5] H. Tanner, G. Pappas, and V. Kumar, “Leader-to-formation stability,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, June 2004.

[6] H. G. Tanner and A. Kumar, “Formation stabilization of multiple agents using decentralized navigation functions,” in *Robotics: Science and Systems I*. Boston: roboticsproceedings.org, June 2005.

[7] G. Antonelli, F. Arrichiello, and S. Chiaverini, “The entrapment/escorting mission,” *Robotics and Automation Magazine, IEEE*, vol. 15, no. 1, pp. 22–29, Mar. 2008.

[8] J. Fenwick, P. Newman, and J. Leonard, “Cooperative concurrent mapping and localization,” in *Proc. 2002 IEEE International Conference on Robotics and Automation*, vol. 2, 2002, pp. 1810–1817.

[9] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, and B. Southall, “A Framework and Architecture for Multi-Robot Coordination,” *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 977–995, 2002.

[10] A. Howard, M. J. Matarić, and G. S. Sukhatme, *Experimental Robotics VIII*. Springer Berlin / Heidelberg, 2003, ch. 12, pp. 146–155.

[11] I. Rekleitis, G. Dudek, and E. Milios, “Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and System, 2002*, vol. 3, 2002, pp. 2690–2695.

[12] S. Roumeliotis and G. Bekey, “Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA’00)*, vol. 3, 2000, pp. 2958–2965.

[13] ——, “Distributed multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[14] R. Negenborn, “Robot Localization and Kalman Filters,” Master’s thesis, Utrecht University, 2003.

[15] S. Thrun, “Simultaneous localization and mapping,” in *Robotics and Cognitive Approaches to Spatial Mapping*. Springer Berlin / Heidelberg, 2008, pp. 13–41.

[16] G. Bradski and A. Kaehler, *Learning OpenCV*. O’Reilly Media Inc., 2008. [Online]. Available: http://oreilly.com/catalog/9780596516130

[17] Various, http://opencv.willowgarage.com.

[18] G. K. Fricke, D. Milutinović, and D. P. Garg, “Sensing and estimation on an experimental testbed for swarm robotics,” in *Proc. 2nd ASME Dynamic Systems and Controls Conference (DSCC’09)*, Oct. 2009.

[19] G. K. Fricke, “Localization, tracking, and odometry calibration of a multi-agent swarm system,” Master’s thesis, Duke University, Department of Mechanical Engineering and Materials Science, Durham, NC, April 2009.

[20] M. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Tehory of NP-Completeness*. W.H. Freeman and Company, 1979.

[21] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The Player/Stage project: Tools for multi-robot and distributed sensor systems,” in *Proc. 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.

[22] D. Eppstein, “Subgraph isomorphism in planar graphs and related problems,” *Journal of Graph Algorithms and Applications*, vol. 3, p. 127, 1999.

[23] A. D. J. Cross, R. C. Wilson, and E. R. Hancock, “Inexact graph matching using genetic search,” *Pattern Recognition*, vol. 30, no. 6, pp. 953 – 970, 1997.

A.2.8 Hazardous Spill Perimeter Detection and Monitoring via Multiple Autonomous Mobile Robitic Agents

The following paper (on the next 8 pages) was co-authored by Guoxian Zhang, Devendra P. Garg, and Gregory K. Fricke, and appeared in the Proceedings of the 2010 ASME Dynamic Systems and Control Conference, Volume 2, on pages 639–646, published and presented in September 2010.

Proceedings of the ASME 2010 Dynamic Systems and Control Conference
DSCC2010
September 12-15, 2010, Cambridge, Massachusetts, USA

DSCC2010-4160

HAZARDOUS SPILL PERIMETER DETECTION AND MONITORING VIA MULTIPLE AUTONOMOUS MOBILE ROBOTIC AGENTS

Guoxian Zhang
Mechanical Engineering & Materials Science Dept.
Duke University
Durham, North Carolina 27708
Email: guoxian.zhang@duke.edu

Devendra P. Garg*
Greg Fricke
Mechanical Engineering & Materials Science Dept.
Duke University
Durham, North Carolina, 27708
Email: {dpgarg, gregory.fricke}@duke.edu

ABSTRACT

The problem of perimeter detection and monitoring has a variety of applications. In this paper, a hybrid system with finite states strategy is proposed for multiple autonomous robotic agents with the purpose of hazardous spill perimeter detection and tracking. In the system, each robotic agent is assumed to be in one of the following states: searching, pursuing, and tracking. The agents are prioritized based on their states, and a potential field is constructed for agents in each state. The location and velocity of an agent in the tracking state as well as those of its closest leading and trailing agents are utilized to control its movement. The convergence of the tracking algorithm is analyzed for multiple spills under certain conditions. Simulation results show that with the proposed method, the agents can successfully detect and track the spills of various sizes and movements.

INTRODUCTION

The development of swarm robotics has emerged as a tool for mobile sensor network in a variety of areas, such as environment monitoring [1], foraging [2] [3], target detection [4], and target tracking [5]. Two different kinds of control, centralized control and decentralized control can be implemented for the robot group. In centralized control, all agents in the group are assumed to be able to share their information [6]. This kind of control suffers from the computational burden for large agent number. More often, decentralized control is used for swarm robotics in which only information within a local network is shared among agents [1] [7]. In this paper we propose a method for the problem of perimeter detection and tracking with swarm robots which may have a variety of applications, for instance: forest fire surveillance [8], oil leakage tracking [9], and animal herd monitoring. Some of the previous work in this area has been reported in literature as follows. Casbeer *et al* [8] proposed a decentralized multiple UAV approach to monitor a forest fire. Each UAV flew along the perimeter of the fire in one direction and then flew backward when it met another UAV. This movement continued and the movement of the UAVs formed a latency. This method could easily adapt to a change in the fire size and the number of UAVs needed during the process. Susca *et al* [10] [11] used a polygon to approximate an environmental boundary. Local information gathered via mobile sensors was used to locate interpolation points to form a polygon. The vertices of the polygon were located uniformly along the boundary. When the boundary was stationary, the method was convergent.

Clark *et al* [9] and Cruz *et al* [12] utilized a hybrid control algorithm to achieve the perimeter search and monitoring in an unknown environment. The robots were initially randomly placed in the work space. When a point on the perimeter was detected by a robot, the robot was navigated by a potential field to approach this point and communicated this information to other robots. When a robot arrived at the perimeter, it was controlled by a tracking controller to travel along the perimeter. Bruemmer *et al* [13] utilized the social potential field generated by different kinds of sensors on each agent to avoid collision and attract agents to the perimeter. The robot learned to respond to fluctuations in sound and light through perception and communication such as sound chirping, touch sensing, and moisture detection.

Another line of research using multiple robots is in the formation control. In this area, a group of agents moves to fulfill some task, such as foraging or tracking, while maintaining

[*] Address all correspondence to this author.

its geometric formation along the way. Olfati-Saber [14] [15] proposed a method to control the movement of a flock. A potential function was designed whose local minima were set to be achieved when the distance between each agent and its neighbors in the flock equalled a desired value. The method was proved to be asymptotically structurally stable under certain conditions. A β agent and a γ agent were proposed to extend this method in an environment having a number of obstacles. The flock could avoid obstacles and then maintain its formation during its movement. The stability of the flocking of a mobile agent group with fixed and dynamic topologies was studied by Tanner *et al* in [16] [17]. A social potential field for the agent group was proposed to control the distance between an agent and its neighbors while the velocity of each agent was controlled by the difference in the velocity from its neighbors. However, in all of these cases, the distance between each pair of agents converged to a desired value which was predetermined. This was not suitable for the case in which the desired distance between two neighbors was unknown before hand or it changed with time.

In this paper, inspired by the flocking of agents, a control law is proposed for a group of agents whose purpose is to search and track a hazardous spill in an unknown environment. Each agent is assumed to have limited and specific sensing and communication ranges. It is shown that the group can successfully detect the spill and track along its boundary when the spill's location and size are changing. Simulation results also showed an excellent performance to track multiple spills that may come together or split during the process.

PROBLEM FORMULATION

A group of autonomous mobile agents $\mathcal{A}_1$, $\mathcal{A}_2$, ..., $\mathcal{A}_n$ is assumed to be initially deployed in the area $\mathcal{W}$. At time t, the configuration of agent i is represented as $\mathbf{q}_i(t) = \{\mathbf{r}_i(t), \theta_i(t)\}$, where $\mathbf{r}_i(t) = \{x_i(t), y_i(t)\}$ denotes the position of $\mathcal{A}_i$ at time t, and $\theta_i(t)$ denotes its orientation. Each agent is assumed to be equipped with a sensor whose field of view (FOV) is $\mathcal{S}_i$, and communication range is $\mathcal{M}_i$. For simplicity's sake, in our paper, we assume that both $\mathcal{S}_i$ and $\mathcal{M}_i$ are circles whose centers are located at $\mathbf{r}_i(t)$, and are of constant radii $r_{\mathcal{S}}$ and $r_{\mathcal{M}}$. This assumption is commonly used by various researchers [14] [18] . A number of spills $\Omega_j(t)$ with the boundary $\partial\Omega_j(t)$ are deployed in $\mathcal{W}$. A spill $\Omega_j(t)$ is defined as a connected subspace in the two dimensional space. Each agent is assumed to be able to measure the length of the spill boundary within its FOV. This length may be computed in different ways. One way is to utilize pattern classification technology to detect the spill boundary from a visual sensor as implemented in [19]. Then the detected spill boundary is represented by a series of points on it and a straight line segment is generated with each pair of adjacent points. The spill boundary length is approximated by summing the lengths of all the straight line segments. When $\partial\Omega_j(t)$ changes, it would be beneficial that the agent group can uniformly distribute along the spill boundary since the change of a part on $\partial\Omega_j(t)$ could be detected by an agent quickly.

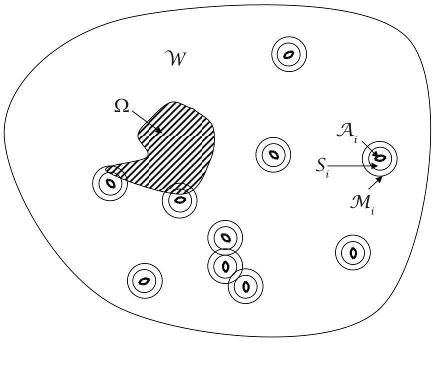


Figure 1. SAMPLE LAYOUT OF THE MULTIPLE AGENT SYSTEM.

A method is proposed to control the movement of each agent tracking a spill. Only the positions of an agent's closest leading and trailing neighbors along its way may be utilized to generate a potential field to control its movement. An influence distance along the spill boundary, denoted by L, controls how the closest neighbor affects an agent's potential field. This assumption guarantees the scalability of proposed method and will be discussed in the "Proposed Method" section.

The problem can be formulated as follows: given a group of agents $\mathcal{A}_1$, $\mathcal{A}_2$, ..., $\mathcal{A}_n$ and their initial configurations $\mathbf{q}_1(0), \mathbf{q}_2(0), \ldots, \mathbf{q}_n(0)$, find the path for each agent i, where $i = 1, 2, \ldots, n$, such that $\forall \varepsilon > 0$ there exists a time $T > 0$ and a spill index j, for $t > T$ agent i tracks $\Omega_j(t)$, and $||\mathbf{r}_i(t), r_{N(\mathbf{r}_i(t))}(t)||_{\partial\Omega_j} = L_j(t)$ satisfies:

$$\begin{cases} |L_j(t) - \frac{||\partial\Omega_j(t)||}{n_j(t)}| < \varepsilon & \text{if} \quad n_j(t) > \frac{||\partial\Omega_j(t)||}{L} \\ L_j(t) >= L - \varepsilon & \text{if} \quad n_j(t) \leq \frac{||\partial\Omega_j(t)||}{L} \end{cases} \tag{1}$$

Here, $||a, b||_{\partial c}$ is the distance from a to b along the curve ∂c with the forward direction as shown in Fig. 2; $N(\mathbf{r}_i(t))$ is the index of the closest neighbor of $\mathcal{A}_i$ in the forward direction; $n_j(t)$ is the number of agents that are tracking Ω_j at time t; and $||\partial\Omega_j(t)||$ is the length of $\partial\Omega_j(t)$. Furthermore, the velocities of agents tracking the same spill converge to the same value. Mathematically speaking, assume that agent i and j are tracking the same spill, $\forall \varepsilon > 0$, $\exists T > 0$, such that $\forall t > T$ we have $|v_i(t) - v_j(t)| < \varepsilon$. When there is no ambiguity, we simplify each variable without writing time; for example, $\partial\Omega_j$ represents $\partial\Omega_j(t)$.

The communication among the mobile agents can be represented as a set of connected undirected graphs, each of which can be defined as $\mathcal{G}_k(\mathcal{V}_k, \mathcal{E}_k)$, where $\mathcal{V}_k = \{a_{l_1}, a_{l_2}, \ldots, a_{l_{\ell k}}\}$ are the agents in $\mathcal{G}_k$, and $\mathcal{E}_k = \{(a_{l_i}, a_{l_j}) \in \mathcal{V}_k \times \mathcal{V}_k \mid a_{l_i}, a_{l_j} \in \mathcal{V}_k, a_{l_j} \in \mathcal{M}_{l_i}\}$. The information, such as a detected point of the spill boundary by an agent, is assumed to be able to be shared within the graph with no time delay. An example of the agent system with ten agents and one spill is shown in Fig. 1.

PROPOSED METHOD

A hybrid control technique is used to control the agents. Three behaviors, searching, pursuing, and tracking, are used for perimeter detection and tracking. The agents are assumed to have a unicycle model in which:

$$\begin{aligned} \dot{x}_i &= v_i cos\theta_i \\ \dot{y}_i &= v_i sin\theta_i \\ \dot{v}_i &= a_i \\ \dot{\theta}_i &= \omega_i \end{aligned} \tag{2}$$

where v_i is the linear velocity of the ith agent, a_i is its linear acceleration, and ω_i is its angular velocity. The control inputs of each agent are set up as $[u_i \;\; w_i]$, where $u_i = a_i$, $w_i = \omega_i$.

Perimeter Searching and Pursuing

In the proposed method, the robotic agents are initially randomly deployed in an unknown environment. The geometry of each agent is ignored in the system, i.e., two agents collide only when their positions are the same. To avoid collision , we assume that there is a repulsive potential field generated by each agent, denoted as U_i [20]:

$$U_i(\mathbf{r}) = \begin{cases} \frac{1}{2}\eta_1(\frac{1}{\rho(\mathbf{r},\mathbf{r}_i)} - \frac{1}{\rho_0})^2 & \text{if} \quad \rho(\mathbf{r},\mathbf{r}_i) \leq \rho_0 \\ 0 & \text{if} \quad \rho(\mathbf{r},\mathbf{r}_i) > \rho_0 \end{cases} \tag{3}$$

where η_1 is a scaling parameter, ρ_0 is the influence distance of each agent, $\mathbf{r}_i$ is the ith agent position, and $\rho(\mathbf{r},\mathbf{r}_i)$ is the distance between $\mathbf{r}$ and $\mathbf{r}_i$.

The agents in different states are assumed to have different priority levels for avoidance. Agents in the searching state are assumed to be at the lowest level, and agents in the tracking state are at the highest level. Let the index sets I_s, I_p, and I_t represent the index of agents in searching, pursuing, and tracking states, respectively. Assume an agent in the searching state is at $\mathbf{r}$, and let $N_s(\mathbf{r}) = \{i \mid \forall i,\ \mathbf{r} \in \mathcal{M}_i\}$ denote the index of agents in whose communication range $\mathbf{r}$ lies. When $N_s(\mathbf{r}) \neq \emptyset$ the following potential field is constructed for the agent:

$$U_s(\mathbf{r}) = \sum_{i \in N_s(\mathbf{r})} U_i(\mathbf{r}) \tag{4}$$

and the force implemented on the agent is computed as:

$$F_s(\mathbf{r}) = -\nabla U_s(\mathbf{r}) \tag{5}$$

$\|F_s(\mathbf{r})\|$ is utilized as control u_i for the agent, and ω_i is set at the value with which the orientation of the agent in the next time step is consistent with the direction of $F_s(\mathbf{r})$. Since the travel distance and search time are not considered in the objective of our problem, when $N_s(\mathbf{r}) = \emptyset$, a random search technique in discrete time with the input $[u_i^r(k), w_i^r(k)]$ is utilized to navigate the agent at $\mathbf{q}$ with searching state:

$$\begin{aligned} u_i^r(k+1) &= u_i^r(k) + b(k)\Delta t \\ w_i^r(k+1) &= w_i^r(k) + c(k)\Delta t \end{aligned} \tag{6}$$

where $b(k)$ is a random variable having uniform distribution with the support assigned by the user, $c(k)$ is a random variable having normal distribution with a zero mean, and variance specified by the user, and Δt is the discrete time period. When other objective such as travel distance is included, collaboration between agents may be used to improve the search efficiency.

When a point lying on the boundary of the spill is within the ith agent sensor's FOV, a detection occurs. The ith agent then changes to a pursuing state and shares the location of the detected point, denoted by $\mathbf{r}_d$, with other agents in its communication graph. A potential field $U_d(\mathbf{r})$ is constructed for the agent moving towards $\mathbf{r}_d$. It is given by:

$$U_d(\mathbf{r}) = \frac{1}{2}\eta_2\rho(\mathbf{r},\mathbf{r}_d)^2 \tag{7}$$

where η_2 is a scaling parameter, and $\rho(\mathbf{r},\mathbf{r}_d)$ is the distance between $\mathbf{r}$ and $\mathbf{r}_d$. Let $N_p(\mathbf{r}) = \{i \mid \forall i \in I_p \cup I_t,\ \mathbf{r} \in \mathcal{M}_i\}$ denote the index of agents in pursuing or tracking state, in whose communication range $\mathbf{r}$ lies. The following potential field is constructed for the agent:

$$U_p(\mathbf{r}) = U_d(\mathbf{r}) + \sum_{i \in N_p(\mathbf{r})} U_i(\mathbf{r}) \tag{8}$$

The negative gradient, $F_p(\mathbf{r})$, of the potential $U_p(\mathbf{r})$ is utilized to navigate the agent in pursuing state. In practice, we assume that there is a maximum velocity, v_{max}, to bound the movement of each agent.

Tracking the Spill Boundary

Assume that the agent i is in the pursuing state. When $\|\mathbf{r}_i - \mathbf{r}_d\| < d$, where d is a capture distance defined by the user, the agent changes into the tracking state, it will choose to move along the spill boundary where the inside of the spill is on its left. The forward direction of the agents' movement is as shown in Fig. 2. When an agent moves away from the forward direction, its velocity is negative. In this case, the unicycle model in Eq. (2) can still be used only by adding π to the orientation. For agent i, l_i is defined as follows. Assume that at time t, the ith agent is tracking Ω_j. It will measure the length of the perimeter and positions of other agents in the tracking state within its FOV. Then, if the closest neighbor at the forward direction, whose index is represented by $N(\mathbf{r}_i)$, is inside its FOV,

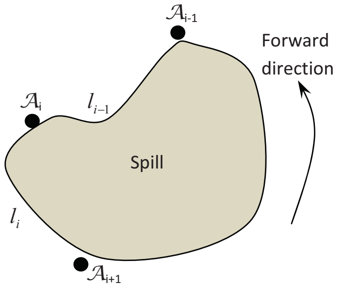


Figure 2. THE LOCATION, FORWARD MOVEMENT DIRECTION, AND DISTANCE BETWEEN ADJACENT MOBILE AGENTS ALONG THE SPILL PERIMETER.

set $l_i = \|\mathbf{r}_i, \mathbf{r}_{N(\mathbf{r}_i)}\|_{\partial\Omega_j}$; otherwise l_i is unknown to the agent. Assume that $\partial\Omega_j$ can be represented by a function $s = f(\mathbf{r})$, where $s \in [0, \|\partial\Omega_j\|]$ indicates the length of the perimeter from a reference point to $\mathbf{r}$ along the forward direction. Write $f(\mathbf{r}_i)$ as s_i and $s(\mathbf{r}_{N(\mathbf{r}_i)})$ as s_{i+1}, then, when l_i is known to agent i, it can be computed as follows:

$$l_i = \begin{cases} s_{i+1} - s_i & \text{if} \quad s_{i+1} \geq s_i \\ s_{i+1} - s_i + \|\partial\Omega_j\| & \text{if} \quad s_{i+1} < s_i \end{cases} \tag{9}$$

Define a virtual distance, L_i between agent i and $N(\mathbf{r}_i)$ as:

$$L_i = \begin{cases} l_i & \text{if} \quad l_i < L \\ L & \text{if} \quad l_i \geq L \text{ or } l_i \text{ is unknown} \end{cases} \tag{10}$$

where L is the influence distance between agent i and agent $N(\mathbf{r}_i)$, and is set as $L = r_S$ in our paper.

Assuming $L_0 = L_{n_j}$, i.e., the last agent wraps to the first agent, a potential function [21] [22] is proposed for agents i, which is given by:

$$V_i(L_i, L_{i-1}) = k_1(2ln(L_i) + \frac{L_{i-1}}{L_i}) \tag{11}$$

where k_1 is a positive constant. When $L_i < L$ and $L_{i-1} < L$,using Eq. (9), Eq. (10) and Eq. (11), the derivative of V_i with respect to s_i is:

$$\frac{dV_i}{ds_i} = -k_1(\frac{1}{L_i} - \frac{L_{i-1}}{L_i^2}) \tag{12}$$

where V_i has a minimum at $L_i = L_{i-1}$. Including the control of velocity for agents, the following controller is proposed to be used as the linear acceleration input of the ith agent.

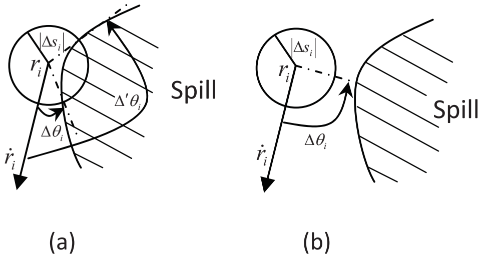


Figure 3. THE CHANGE IN θ_i AT EACH TIME STEP. a) THE MOVEMENT OF AGENT AND SPILL HAVE INTERSECTION, IF $\Delta s_i > 0$, ANGULAR CHANGE IS $\Delta\theta_i$; IF $\Delta s_i < 0$, ANGULAR CHANGE IS $\Delta'\theta_i$; b) THE MOVEMENT OF AGENT AND SPILL HAVE NO INTERSECTION, AGENT MOVES TOWARDS THE CLOSEST BOUNDARY POINT OF THE SPILL.

$$u_i = \begin{cases} k_1(\frac{1}{L_i} - \frac{L_{i-1}}{L_i^2}) + (v_{i+1} - 2v_i + v_{i-1}) & \text{if } L_i < L \text{ , } L_{i-1} < L \\ k_1(\frac{1}{L} - \frac{L_{i-1}}{L^2}) + (v_{i-1} - v_i) & \text{if } L_i = L \text{ , } L_{i-1} < L \\ k_1(\frac{1}{L_i} - \frac{L}{L_i^2}) + (v_{i+1} - v_i) & \text{if } L_i < L \text{ , } L_{i-1} = L \\ 0 & \text{if } L_i = L \text{ , } L_{i-1} = L \end{cases} \tag{13}$$

From Eq. (12) and Eq. (13) it is evident that when L_i goes to zero, u_i goes to minus infinity. This guarantees that the ith agent does not surpass the $(i+1)$th agent, which implies that the order of the agents is maintained when they are moving.

The angular velocity of the ith agent is also controlled discretely. Assume that the change in distance traveled by the ith agent during Δt is Δs_i. The change in θ_i, denoted as $\Delta\theta$, is computed in different situations as shown in Fig. 3.

If another kind of agent, denoted as $\mathcal{B}$, which has a much larger measurement range, (e.g., an airborne agent) is available to monitor the spill when the multiple robotic agents are detecting the spill, its measurement can be used as a coarse measure of l_i. This measurement of l_i usually has a lot of noise due to the low resolution of the global measurement. In that case, Eq. (10) may be changed to:

$$L_i = \begin{cases} l_i, & \text{if} \quad l_i < L \\ \hat{l}_i, & \text{if} \quad l_i \geq L \end{cases} \tag{14}$$

where $\hat{l}_i$ is the noisy measurement by agent $\mathcal{B}$. This case will be studied in our future work.

Stability Analysis

To investigate the system's stability, we consider a time $t > T$ in which we assume that all agents are in the tracking state, and the spills stop changing in size and location. Another assumption is that the number of agents is large enough so that each agent is connected with its leading and trailing neighbors during tracking. Without loss of generality, we relabel the agents so that the first n_1 agents track Ω_1, the (n_1+1)th to (n_1+n_2)th agents track Ω_2, and so forth. Let $\mathbf{l} = [L_1, L_2, \ldots, L_n]^T$. The collective potential of the system can be given by:

$$V(\mathbf{s}) = V(\mathbf{l}) = \sum_{j=1}^{p} \left(\sum_{i=n_0+\ldots+n_{j-1}+1}^{n_0+\ldots+n_j} V_i \right) \tag{15}$$

where $\mathbf{s} = [s_1, \ldots, s_n]^T$; p is the number of spills when the spills stop changing; n_0 equals zero; and with $M_1, M_2, \ldots, M_p$ being the lengths of the p spill boundaries, $L_1, L_2, \ldots, L_n$ satisfies:

$$\begin{cases} L_1 + L_2 + \ldots + L_{n_1} = M_1 \\ L_{n_1+1} + L_{n_1+2} + \ldots + L_{n_1+n_2} = M_2 \\ \ldots \\ L_{n_1+\ldots+n_{p-1}+1} + L_{n_1+\ldots+n_{p-1}+2} + \ldots + L_n = M_p \end{cases} \tag{16}$$

The collective dynamics of the system is given by:

$$\dot{\mathbf{s}} = \mathbf{v} \tag{17}$$

$$\dot{\mathbf{v}} = -[\frac{\partial V_1}{\partial s_1}, \frac{\partial V_2}{\partial s_2}, \ldots, \frac{\partial V_n}{\partial s_n}]^T - \mathcal{L}(\mathbf{s})\mathbf{v} \tag{18}$$

where $\mathbf{v} = [v_1, \ldots, v_n]^T$, and $\mathcal{L}$ is the graph Laplacian as described in references [14] and [23].

Using Lagrange Multipliers, Eq. (15) has an extreme point at

$$\begin{cases} L_1 = L_2 = \ldots = L_{n_1} = \frac{M_1}{n_1} \\ L_{n_1+1} = L_{n_1+2} = \ldots = L_{n_1+n_2} = \frac{M_2}{n_2} \\ \ldots \\ L_{n_1+\ldots+n_{p-1}+1} = L_{n_1+\ldots+n_{p-1}+2} = \ldots = L_n = \frac{M_p}{n_p} \end{cases} \tag{19}$$

Although it is analytically difficult to show that this point is the only minimum for Eq. (15), numerical simulations with matlab support this assumption. For example, when $n = 3$, $M = 24$, $p = 1$, and $k = 1$, the value of V is shown in Fig. 4, where the minimum of V appears at $L_1 = L_2 = L_3 = 8$ and satisfies Eq. (19). Eq. (19) is satisfied for other values of n, M, and p also as verified by simulations.

Let $H(\mathbf{s}, \mathbf{v})$ represent the energy of the system, i.e.,

$$H(\mathbf{s}, \mathbf{v}) = V(\mathbf{s}) + \frac{1}{2}\sum_{i=1}^{n} v_i^2 \tag{20}$$

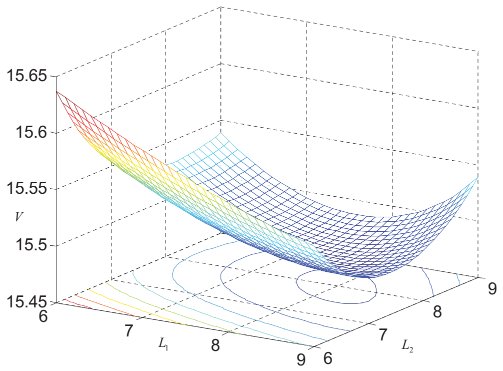


Figure 4. AN EXAMPLE OF THE COLLECTIVE POTENTIAL VALUE CHANGING WITH L_1 AND L_2.

Then

$$\begin{aligned} \dot{H}(\mathbf{s}, \mathbf{v}) =& \nabla V(\mathbf{s})^T \mathbf{v} - [\frac{\partial V_1}{\partial s_1}, \ldots, \frac{\partial V_n}{\partial s_n}]^T \mathbf{v} - \mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v} \\ =& [\frac{\partial V_{n_1}}{\partial s_1} + \frac{\partial V_2}{\partial s_1}, \ldots, \frac{\partial V_{n-1}}{\partial s_n} + \frac{\partial V_{n-n_p+1}}{\partial s_n}]\mathbf{v} - \mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v} \\ =& \left\{ [\frac{2}{L_{n_1}} - \frac{L_{n_1-1}}{L_{n_1}^2} - \frac{1}{L_2}, \ldots, \frac{2}{L_{n-1}} - \frac{L_{n-2}}{L_{n-1}^2} - \frac{1}{L_{n-n_p+1}}]\mathbf{v} \right\} \\ & - \mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v} \end{aligned} \tag{21}$$

where $\mathcal{L}$ has the form

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_1 & & & \\ & \mathcal{L}_2 & & \\ & & \ddots & \\ & & & \mathcal{L}_p \end{pmatrix} \tag{22}$$

which is a positive semidefinite block diagonal matrix with rank $n-p$, and $\mathcal{L}_j$ has a dimension n_j. In general, $\dot{H}$ may be positive, this is caused by the case that only information of the closest leading and trailing neighbors is utilized to generate the potential function for agent i. Therefore, for agent i, no information on V_{i-1} and V_{i+1} is available. However, with bounded $\mathbf{v}$ and the control in Eq. (13), each L_i, where $i = 1, 2, \ldots, n$, tends to be close to each other, and then the first set of terms in curly brackets on the right side of Eq. (21) gets close to zero which makes the sign of $\dot{H}$ dominated by $-\mathbf{v}^T \mathcal{L}(\mathbf{s})\mathbf{v}$ and be negative.

In this case, $\dot{H}$ equals zero when $\mathbf{v}$ is an eigenvector of $\mathcal{L}$ corresponding to eigenvalue zero. Then $\mathbf{v}$ will be in the space with basis containing p vectors, denoted by $\mathbf{b}_i = [c_1, c_2, \ldots, c_n]$, where

$$c_j = \begin{cases} 1 & n_0 + \ldots + n_{i-1} + 1 < j \leq n_0 + \ldots + n_i \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

for $i = 1, 2, \dots, p$, and

$$\mathbf{v} = \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 + \dots + \lambda_p \mathbf{b}_p \tag{24}$$

where λ_i, $i = 1, 2, \dots, p$, is an arbitrary parameter, which in this case represents the steady linear velocity of all agents on $\partial\Omega_i$. This shows that the velocities of all agents tracking the same spill converge to the same value. Since $\mathbf{v}$ is bounded, the convergent velocity is also bounded. When H achieves a minimum, V is also at its local minimum. From previous analysis,

$$L_{n_1+\ldots+n_{j-1}+1} = \ldots = L_{n_1+\ldots+n_j}, \quad \text{for } j = 1, 2, \dots, p \tag{25}$$

is a local minimum for V. Hence, the system achieves the stable state with equal distance between adjacent agents and equal velocity for all agents tracking the same spill.

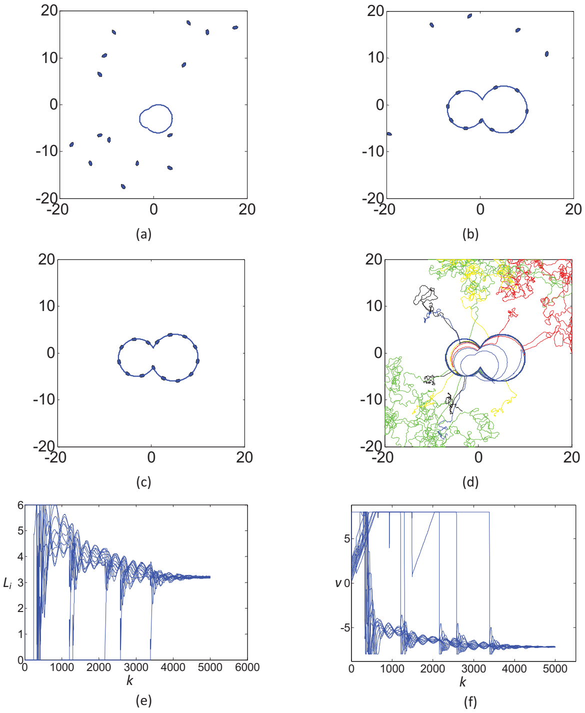
Consider a case that after all agents change to tracking state, spill Ω_1 does not have enough agents tracking along its boundary, i.e., $n_1 < \frac{\|\partial\Omega_1\|}{L}$. We claim that all agents tracking Ω_1 converge to the same velocity and the distance between each pair of adjacent agents converges to a value no less than L. The energy of the agents tracking Ω_1 is presented as:

$$H_1(\mathbf{s}_1, \mathbf{v}_1) = V(\mathbf{s}_1) + \frac{1}{2}\sum_{i=1}^{n_1} v_i^2 \tag{26}$$

Similarly:

$$\begin{aligned}\dot{H}(\mathbf{s}_1, \mathbf{v}_1) =& [\frac{2}{L_{n_1}} - \frac{L_{n_1-1}}{L_{n_1}^2} - \frac{1}{L_2}, \dots, \frac{2}{L_{n_1-1}} - \frac{L_{n_1-2}}{L_{n_1-1}^2} - \frac{1}{L_1}]\mathbf{v}_1 \\ & - \mathbf{v}_1^T \mathcal{L}(\mathbf{s}_1)\mathbf{v}_1 \end{aligned} \tag{27}$$

where $\mathbf{s}_1 = [s_1, \dots, s_{n_1}]$ and $\mathbf{v}_1 = [v_1, \dots, v_{n_1}]$. Without loss of generality, we assume that at time t, $\mathcal{L}_1$ is a block diagonal matrix containing two blocks $\mathcal{L}_{11}$ and $\mathcal{L}_{12}$. In view of the discussion above, if no connection between $\mathcal{L}_{11}$ and $\mathcal{L}_{12}$ holds, the velocity of agents in $\mathcal{L}_{11}$ converges to v^1, and velocity of agents in $\mathcal{L}_{12}$ converges to v^2. If $v^1 \neq v^2$, the two connected graphs will connect with each other some time since they are tracking a perimeter. Then Eq. (27) is likely to be negative and cause the energy of the system to decrease again until either the graph splits into several connected graphs or the velocity of all agents converges to the same value. Since different velocities of agents in various connected graphs cause a decrease of the system energy, the system energy achieves a local minimum when the agents within all connected graphs tracking Ω_1 have the same velocity. In this case, the first agent at the forward direction of each connected graph will have a leading virtual distance equal to L (i.e., the distance between this agent and its leading adjacent agent is larger than L), which causes the distance between adjacent agents within its graph to be L to have a minimal energy also. Therefore, the distance between each pair of adjacent agents tracking Ω_1 is not less than L.



(a) (b) (c) (d) (e) (f)

Figure 5. SIMULATION RESULT WITH 15 AGENTS AND ONE SPILL. (a) SNAP GRAPH AT TIME STEP ONE; (b) SNAP GRAPH AT TIME STEP 600; (c) SNAP GRAPH AT TIME STEP 5000; (d) TRAJECTORY OF AGENTS; (e) DISTANCE BETWEEN EACH AGENT AND THE CLOSEST AGENT AHEAD OF IT; (f) THE VELOCITY OF EACH AGENT.

SIMULATION AND RESULTS

In this section, simulations are described to test the performance of the proposed method based on different conditions. The influence of spill locations, size and numbers as well as the agent group size are studied via a number of simulations.

Sensitivity to Changes in Size and Location of Spill

In this subsection, we design a simulation to see how the agent group performs when the spill's location and size can change. We will show that our method can search and track the spill successfully. We assume that 15 agents are randomly distributed in the environment at the beginning. $\mathcal{W}$ is a square arena with sides of length 40 meters, $r_S = 6m$, $r_{\mathcal{M}} = 8m$, $v_{max} = 8m/s$, and $\Delta t = 0.02s$. The results of one run are shown in Fig. 5. From the results, especially 5(d), we can see that the agents in the tracking state can follow the change in the location and size of the spill. We assume that for each agent i, before changing to the tracking state, L_i equals zero. Therefore, in Fig. 5(e) each jump

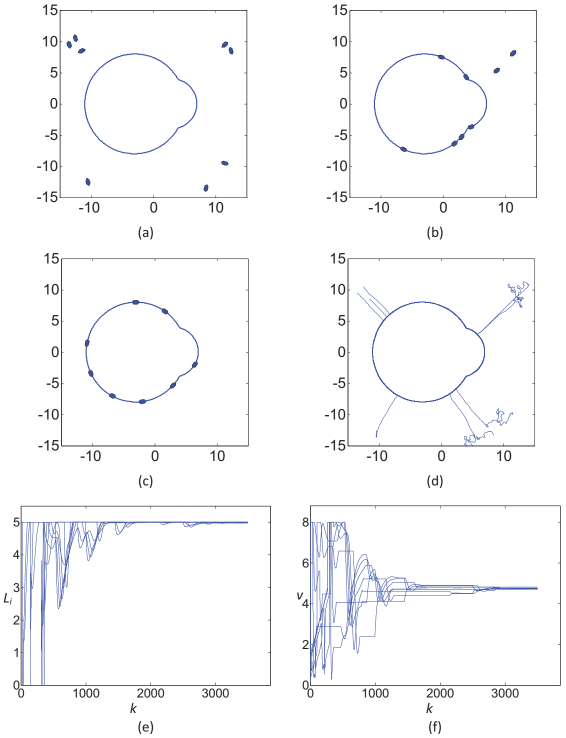


Figure 6. SIMULATION RESULT WITH 8 AGENTS AND ONE SPILL. (a) SNAP GRAPH AT TIME STEP ONE; (b) SNAP GRAPH AT TIME STEP 320; (c) SNAP GRAPH AT TIME STEP 3500; (d) TRAJECTORY OF AGENTS; (e) DISTANCE BETWEEN EACH AGENT AND THE CLOSEST AGENT AHEAD OF IT; (f) THE VELOCITY OF EACH AGENT.

of L_i from zero indicates that one agent changes to the tracking state. As shown in Fig. 5(e) and Fig. 5(f) during the process, when an agent arrives at the spill and changes to the tracking state, it causes change in the velocity and virtual distance of the group, and after that the agent's new group starts to converge to a new stable state. After all the agents are in the tracking state, the whole agent group converges to the same velocity and with the same distance between adjacent agents.

Sensitivity to Number of Agents

In this subsection, we assume the spill's location, shape and size are fixed, and the purpose is to test how the proposed method performs when there are not enough agents in the environment. In this simulation, we assume that $\mathcal{W}$ is a square arena with sides of length 30 meters, $r_{\mathcal{S}} = 5m$, $r_{\mathcal{M}} = 8m$, $v_{max} = 8m/s$, and $\Delta t = 0.02s$. The results of one run are shown in Fig. 6. From the results we can see that the number of agents is not enough to keep all agents in one connected graph when tracking the spill. As time goes on, all the agents change to the tracking state, the virtual distance of each agent converges to $L = 5m$, and their velocity converges to about $4.5m/s$. From Fig. 6(c) we can see that the agents form two connected graphs with the distance between adjacent agents in each graph equalling L. It can be seen in Fig. 6(e) that at about the 2600th time step, there is a slight variation at L_i. It may be caused by the discretization of the time in simulation.

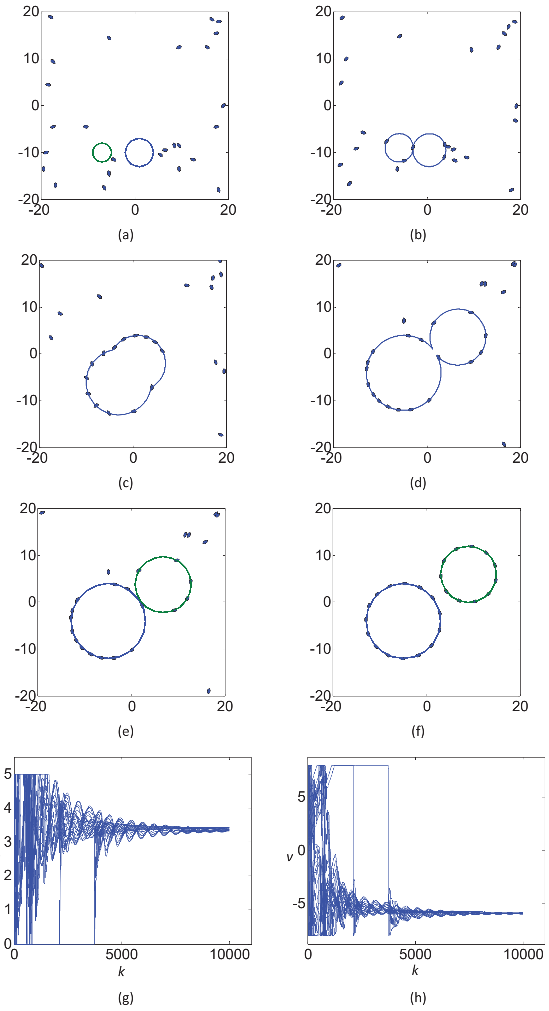


Figure 7. SIMULATION RESULT WITH CHANGING SPILL NUMBER. (a) SNAP GRAPH AT TIME STEP ONE; (b) SNAP GRAPH AT TIME STEP 52; (c) SNAP GRAPH AT TIME STEP 400; (d) SNAP GRAPH AT TIME STEP 682; (e) SNAP GRAPH AT TIME STEP 690; (f) SNAP GRAPH AT TIME STEP 5000; (g) DISTANCE BETWEEN EACH AGENT AND THE CLOSEST AGENT AHEAD OF IT; (h) THE VELOCITY OF EACH AGENT.

Sensitivity to Change in the Number of Spills

In this subsection, we build a simulation to see how the agent group adapts the change of the spill number. In this environment, we assume that there are two spills at the beginning. As time goes on the two spills merge into one spill, and then again split into two spills. It is assumed that the number of agents is 26, $\mathcal{W}$ is a square arena with sides of length 40 meters, $r_{\mathcal{S}} = 5m$, $r_{\mathcal{M}} = 8m$, $v_{max} = 8m/s$, and $\Delta t = 0.01s$. The results are shown in Fig. 7.

From the snaps of the simulation process, we can see that the agents successfully search, pursue and track the spills no matter if they are merged or they split into pieces. After the spills stop changing, the velocities of agents tracking the same spill converge to the same value as well as maintaining constant distance between adjacent agents.

CONCLUSIONS

In this paper, a hybrid system with feedback control strategy is advanced for a group of agents with a limited sensor field of view (FOV) and communication range to search and track a number of spills in an unknown environment. A hierarchical potential field is designed for agents in different states to control their movement. Simulation results show that the agents can successfully search and track spills whose location, shape, size, and number may change over time.

ACKNOWLEDGMENT

The research reported in this paper was supported by the U.S. Army Research Office under grant number W911NF-08-0106.

REFERENCES

[1] Sahin, E., ed., 2005. *Swarm Robotics*, Vol. 3342 of *Lecture Notes in Computer Sciencee*. Springer Berlin, Heidelberg, Germany, Chap. 2, pp. 10–20.

[2] Labella, T. H., and Dorigo, M., 2006. “Division of labor in a group of robots inspired by ants’ foraging behavior”. *ACM Transactions on Autonomous and Adaptive Systems,* **1**(1), September, pp. 4–25.

[3] Zhang, G., and Garg, D., 2009. “Mobile multi-robot control in target search and retrieval”. In Proc. of the 1st Annual Dynamic Systems and Control (DSC) Conference.

[4] Bourgault, F., Goktogan, A., Furukawa, T., and Durrant-Whyte, H., 2004. “Coordinated search for a lost target in a Bayesian world”. *Advanced Robotics,* **18**, pp. 979–1000.

[5] Shucker, B., and Bennett, J. K., 2005. “Target tracking with distributed robotic macrosensors”. In Proceedings of MILCOM 2005.

[6] Schumacher, C., Chandler, P., and Rasmussen, S., 2002. “Task allocation for wide area search munitions via iterative network flow”. In AIAA Guidance, Navigation, and Control Conference and Exhibit.

[7] Alanyali, M., Venkatesh, S., Savas, O., and Aeron, S., 2004. “Distributed Bayesian hypothesis testing in sensor networks”. In Proceedings of American Control Conference, Vol. 6, pp. 5369–5374.

[8] Casbeer, D., Kingston, D., Beard, R., Mclain, T., Li, S., and Mehra, R., 2006. “Cooperative forest fire surveillance using a team of small unmanned air vehicles”. *International Journal of Systems Science,* **37**(6), pp. 351–360.

[9] Clark, J., and Fierro, R., 2007. “Mobile robotic sensors for perimeter detection and tracking”. *ISA Transactions,* **46**, pp. 3–13.

[10] Susca, S., Bullo, F., and Martinez, S., 2009. “Gradient algorithms for polygonal approximation of convex contours”. *Automatica,* **45**, pp. 510–516.

[11] Susca, S., Martínez, S., and Bullo, F., 2008. “Monitoring environmental boundaries with a robotic sensor network”. *IEEE Transactions on Control Systems Technology,* **16**(2), pp. 288–296.

[12] Cruz, D., McClintock, J., Perteet, B., Orqueda, O., Cao, Y., and Fierro, R., 2007. “Decentralized cooperative control - a multivehicle platform for research in networked embedded systems”. *IEEE Control Systems Magazine,* **27**, pp. 58–78.

[13] Bruemmer, D., Dudenhoeffer, D., Anderson, M., and McKay, M., 2002. “A robotic swarm for spill finding and perimeter formation”. *Spectrum.*

[14] Olfati-Saber, R., 2006. “Flocking for multi-agent dynamic systems: Algorithms and theory”. *IEEE Transactions on Automatic Control,* **51**(3), pp. 401–420.

[15] Olfati-Saber, R., and Murray, R., 2003. “Flocking with obstacle avoidance: cooperation with limited communication in mobile networks”. In Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 2022–2028.

[16] Tanner, H., Jadbabaie, A., and Pappas, G., 2003. “Stable flocking of mobile agents, part I: Fixed topology”. In Proceedings of the IEEE Conference on Decision and Control, pp. 2010–2015.

[17] Tanner, H., Jadbabaie, A., and Pappas, G., 2003. “Stable flocking of mobile agents, part II: Dynamic topology”. In Proceedings of the IEEE Conference on Decision and Control, pp. 2016–2021.

[18] Yang, Z., Zhang, Q., and Chen, Z., 2000. “Choosing good distance metrics and local planners for probabilistic roadmap methods”. *IEEE Transactions on Robotics and Automation,* **16**(4).

[19] Fricke, G., Zhang, G., Caccavale, A., Li, W., and Garg, D., September 2010. “An intelligent sensing network of distributed swarming agents for perimeter detection and surveillance”. In Proceedings of ASME Dynamic Systems and Control Conference (DSCC2010).

[20] Latombe, J. C., 1991. *Robot Motion Planning*. Kluwer Academic Publishers.

[21] Leonard, N., and Fiorelli, E., 2001. “Virtual leaders, artificial potentials and coordinated control of groups”. In Proceedings of the IEEE International Conference on Decision and Control, pp. 2968–2973.

[22] Kumar, M., Garg, D., and Kumar, V., 2010. “Segregation of heterogeneous units in a swarm of robotic agents”. *IEEE Transactions on Automatic Control,* **55**(3), pp. 743–748.

[23] Luxburg, U., 2007. “A tutorial on spectral clustering”. *Statistics and Computing,* **17**(4), pp. 395–416.

A.2.9 An Intelligent Sensing Network of Distributed Swarming Agents for Perimeter Detection and Surveillance

The following paper (on the next 8 pages) was co-authored by Gregory K. Fricke, Guoxian Zhang, Adam W. Caccavale, S. Walter Li, and Devendra P. Garg, and appeared in the Proceedings of the 2010 ASME Dynamic Systems and Control Conference, Volume 2, on pages 741–748, published and presented in September 2010.

Proceedings of the ASME 2010 Dynamic Systems and Control Conference
DSCC2010
September 12-15, 2010, Cambridge, Massachusetts, USA

DSCC2010-4256

AN INTELLIGENT SENSING NETWORK OF DISTRIBUTED SWARMING AGENTS FOR PERIMETER DETECTION AND SURVEILLANCE

Gregory K. Fricke, Guoxian Zhang[*]
Adam Caccavale, Walter Li[†]
Devendra P. Garg[‡]
Robotics and Manufacturing Automation Laboratory
Mechanical Engineering and Materials Science Department
Duke University
{gregory.fricke, guoxian.zhang, adam.caccavale, walter.li, dpgarg}@duke.edu

ABSTRACT

Perimeter detection and surveillance by a distributed swarm of autonomous agents offers a solution in multiple cases where human safety or environmental protection is of concern. An algorithmic method is developed for detection and surveillance of one or more dynamic perimeters via distributed control under local-area sensing and communication. Local graphs among neighbors are formed based on proximity, at which point communication among graph members becomes possible. Hardware-in-the-loop mixed-simulation experiments validate the method utilizing iRobot Create® ground robots and a NaturalPoint OptiTrack vision system.

NOMENCLATURE

$a(t)$	Linear acceleration control input at time t
$\omega(t)$	Angular velocity control input at time t
$X(t)$	Dynamic state vector for the ground robots
I_s	Set of indices of agents in Search state
I_p	Set of indices of agents in Pursuit state
I_t	Set of indices of agents in Track state
N	Total number of agents
$N_s(\mathbf{r})$	Set of agents influencing Search-state agent at $\mathbf{r}$
$N_p(\mathbf{r})$	Set of agents influencing Pursuit-state agent at $\mathbf{r}$
$N_t(\mathbf{r})$	Set of agents influencing Track-state agent at $\mathbf{r}$
$U_i(\mathbf{r})$	Repulsive potential field at $\mathbf{r}$ due to robot i
$U_s(\mathbf{r})$	Repulsive potential field by a Search-state agent
$U_p(\mathbf{r})$	Repulsive potential field by a Pursuit-state agent
$U_t(\mathbf{r})$	Potential field by a Track-state agent
$U(\mathbf{r})$	Total potential field acting on an agent at $\mathbf{r}$
$\mathbf{r}$	2D coordinates on the plane
$\rho(.,.)$	Euclidean distance between two coordinates
ρ_0	Influence range for repulsive potential functions
ρ_c	Maximum communication range for robots
ρ_s	Range at which robots can sense spills
ρ_r	Range at which robots can sense each other
k	Discrete-time sample index
η_s, η_p, η_t	Shaping constants for potential functions

INTRODUCTION

A primary motivation for the application of robotic technology is to replace humans in hazardous or repetitive conditions. Such applications include monitoring the perimeter of forest fires and land- and waterway-based hazardous material spills. Due to the hazardous conditions present, fully-distributed control of independent-but-cooperative swarming agents offers an ideal fit in view of its inherent redundancy and graceful degradation [1]. Detection and localization of hazardous materials has been at the forefront of robotic exploration research for many years (see, e.g., [2] and included citations). Forest fire perimeter monitoring by UAVs was explored by Casbeer, Kingston et al [3, 4].

Additional perimeter detection and monitoring research is

[*] Mr. Fricke is a current PhD student and Mr. Zhang is a current PhD candidate in the Pratt School of Engineering at Duke University. Both are student members of the ASME - DSC Division.

[†] Mr. Caccavale and Mr. Li are current undergraduate students and Pratt Fellows in the Pratt School of Engineering at Duke University.

[‡] Professor Garg is with the Department of Mechanical Engineering and Materials Science of the Pratt School of Engineering at Duke University and is a Life Fellow of the ASME.

reported by Clark *et al* [5], Cruz *et al* [6], and Bruemmer *et al* [7]. Each of these works make use of potential functions for robotic navigation during some part of the algorithm.

Recent simulation-based research has been conducted [8] in the context of perimeter tracking and surveillance under the condition that a perimeter may be changing or even splitting into multiple distinct areas. The work detailed in this paper is focused on complementary experimental validation of the perimeter tracking method developed in that recent work.

Many simplifying assumptions are made in the conduct of simulations. For example, sensor fields of view (FOVs) are assumed to be perfect circles with no change in performance with range. Communication is assumed to be perfect and instantaneous. The robots are assumed to be point-sized, and collisions only occur if robots exactly overlap one another. Robot control may use a unicycle model taking into account robot heading, but changes in heading – regardless of magnitude – are accomplished in a single time-step with no error.

Problem Statement

The goal of this work is to quantify the performance of N autonomous ground vehicles $\mathcal{A}_1, \ldots, \mathcal{A}_N$ performing perimeter detection and surveillance in a confined but unknown planar region $\mathcal{W}$. A set of robots will be distributed in an *a priori* unknown environment containing at least one region Ω with boundary $\delta\Omega$. The starting configurations of the robots ($\mathbf{q}_i = [x_i,\ y_i,\ \theta_i]^T$) are randomized. The robots should explore the environment in search of the region. Upon detection, the boundary should be followed. When multiple robots are following the same boundary, they should converge to the same linear velocity and exhibit constant spacing between any pair of robots. If these conditions are met, the experimental mission shall be marked successful.

Effects of various parameters on the swarm performance are investigated. The parameters of interest in this case are the sensing and communication ranges between robots (which are simulated and can be easily changed for different experiment instances and on a per robot basis) and the number of robots participating in the task. The success rate and time-to-completion under these parametric variations should be evaluated for notable trends.

APPROACH

Control of the robots is accomplished via fully distributed sensing and computation. The only explicit cooperation is under the sharing of information within a robot's local connectivity graph. The information yielded under this architecture of limited communication may be considered another type of sensing.

Control Laws

The ground robots used in this experiment are differential-drive robots with a front-castor wheel. The dynamic model for such robots, frequently called *unicycle* robots, is given in Eq. (1).

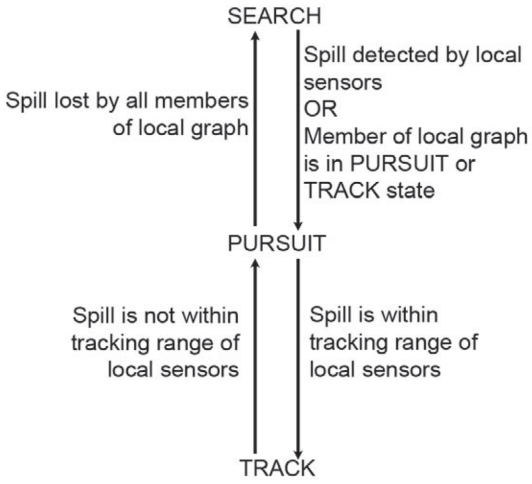

SEARCH
Spill lost by all members of local graph
Spill detected by local sensors OR Member of local graph is in PURSUIT or TRACK state
PURSUIT
Spill is not within tracking range of local sensors
Spill is within tracking range of local sensors
TRACK

Figure 1. STATE-TRANSITION DIAGRAM.

The control inputs, $a(t)$ and $\omega(t)$, represent the linear acceleration and angular velocity, respectively.

$$\dot{X}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a(t) \\ \omega(t) \end{bmatrix} \qquad (1)$$

The control law applied to the robot (e.g., the construction of $a(t)$ and $\omega(t)$) depends on the robot's individual state. The control modes are Search, Pursuit, and Track. In the software implementation, these modes are extended with Idle and Error states for practical considerations. The algorithmic state transitions for an individual robot are shown in Fig. 1.

Algorithmically, the three control-oriented states are identical in the context of which functionalities are executed. However, the data processing and control laws are distinct for each state. In all states, collision avoidance between robots must be ensured. Thus, in all control states, a repulsive potential field (see, e.g., [9–11]) is generated by each robot. The extent to which a robot is affected by the generated potential fields depends on the priority level of the robots, which is defined by the robot's current state. Robots in Track state have the highest priority, and thus will only be repulsed by other robots in Track state. Robots in Search state have the lowest priority, meaning that they will be repulsed by all robots. Robots in Pursuit state are repulsed by others in Pursuit or Track state, but not by those in Search state.

For agent i at location $\mathbf{r}$, the repulsive field generated is given in Eq. (2). The function $\rho(.\ ,\ .)$ is a vector difference function, and ρ_0 is the range of influence of the repelling field.

$$U_i(\mathbf{r}) = \begin{cases} \frac{1}{2}\eta_s\left(\frac{1}{\rho(\mathbf{r},\mathbf{r}_i)} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho(\mathbf{r},\mathbf{r}_i) \le \rho_0 \\ 0 & \text{if } \rho(\mathbf{r},\mathbf{r}_i) > \rho_0 \end{cases} \qquad (2)$$

Search State A robot that is in Search state, but has not detected a spill or any other robot, will explore its environment according to a stochastic navigation process. For a given agent j at $\mathbf{r}_j$ in Search state, the set of agents within communication range ρ_c is given by $N_s(\mathbf{r}_j) = \{i|\forall i, \rho(\mathbf{r}_j, \mathbf{r}_i) < \rho_c, i \neq j\}$.

When $N_s(\mathbf{r}_j) = \emptyset$, the navigation control is a random excitement given in Eq. (3). The random variables b and c are defined by the user, and are uniformly distributed, respectively. The center of the uniform distribution for b is slightly positive, resulting in forward motion over time, whereas for c the center is 0 such that there is no preferential direction for the robots. Note that these control variables in discrete time map to the continuous control parameters defined earlier, i.e., $u(k) \sim a(t)$ and $w(k) \sim \omega(t)$.

$$\begin{aligned} u_i(k+1) &= u_i(k) + b(k)\Delta t \\ w_i(k+1) &= w_i(k) + c(k)\Delta t \end{aligned} \tag{3}$$

Important sensing events relative to the search task includes the detection of the spill or detection of a neighbor. If another robot is detected, $N_s(\mathbf{r}_j)$ becomes non-empty and an obstacle avoidance component is added to the control.

The total repulsive field experienced by agent j for $N_s \neq \emptyset$ is thus the sum of the fields of the agents within range, as shown in Eq. (4).

$$U(\mathbf{r}_j) = \sum_{i \in N_s} U_i(\mathbf{r}_j) \tag{4}$$

The negative gradient of this summed field, $F_j = -\nabla U(\mathbf{r}_j)$ yields the collision-avoidance control input, where

$$u_j(k) = ||F_j(k)|| \tag{5}$$

$$w(k) = \frac{\theta_j(k) - \angle(F_j(k))}{\Delta t} \tag{6}$$

For $N_s(\mathbf{r}_j) = \emptyset$, the control is governed by the controller specific to the state. In Search state, the control is dictated by neighbor separation: either the control is completely random, or is dictated by the proximity of another robot.

If a spill is detected by the robot or another member of the local graph, the robot will transition to Pursuit state using the detected point, $\mathbf{r}_d$, as a navigational goal.

Pursuit State In Pursuit state, the robot acts on some information to guide it to the spill. This information may be a spill-detection event, wherein a point within the robot's sensing range is identified as part of a spill. Alternatively, a member of the robot's local communication graph may have entered Pursuit (or Track) state, indicating communicable knowledge of a spill. The detected point, $\mathbf{r}_d$, is made available to all members of the local graph. The control law to navigate the robot to this detection point is given in Eq. (7).

$$U_d(\mathbf{r}) = \frac{1}{2}\eta_p(\rho(\mathbf{r}, \mathbf{r}_d))^2 \tag{7}$$

Using only this control law would likely cause frequent collisions, as it is possible that multiple robots would receive the same detected point, $\mathbf{r}_d$, as a goal. As those robots converge on that point, collisions could be difficult to avoid. Thus collision avoidance very similar to that of Eq. (2) and Eq. (4) is calculated, except that the summation is taken over the set N_p to yield potential U_p. The set of neighbors that influence a robot in Pursuit state at coordinates $\mathbf{r}$ is given by $N_p(\mathbf{r}) = \{i|\forall i \in I_p \cup I_t, \rho(\mathbf{r}, \mathbf{r}_i) < r_c\}$. The total potential field is then given by $U(\mathbf{r}) = U_d(\mathbf{r}) + U_p(\mathbf{r})$, and the negative gradient $F_p(\mathbf{r}) = -\nabla U(\mathbf{r})$ defines the control input just as in Eq. (5) and Eq. (6).

In [8], the transition from Pursuit to Track state occurs when the robot is sufficiently close to the spill. In this experimental case the robots are limited to the downward looking IR sensors mounted on their front bumpers, implying that the robot cannot detect the spill until it is already sufficiently close to enter Track state. Thus the time spent in Pursuit state in these experiments will be only one time-step if the robot has directly detected the spill. The cases where a robot remains in Pursuit state for extended time are when spill detection occurs by another member of the local communication graph.

If the robots had cameras or forward-looking-IR sensors, their spill-sensing range could be extended to more closely mimic the behavior simulated in [8], although the FOV in these cases would be limited to the region in front of the robot rather than a circular region around the robot. Alternatively an aerial observer could be utilized to provide coarse detection information regarding the spill to broadcast to the distributed robots, guiding their search. Both of these situations are discussed later in this paper.

Track State The Track state controller is engaged when the robot is sufficiently close to the spill perimeter that its own sensors can closely follow the edge. Once in Track state, the heading will thus be determined by the curvature of the perimeter to be tracked. The problem of achieving balanced and distributed coverage of the entire perimeter leads to a different potential function for control. The control parameters of interest in Track state are the distances, ℓ_i, ℓ_{i-1}, to the nearest leading and trailing neighbors *along the perimeter*, as well as the velocities of the robot and those neighbors.

Defining an influence distance of $L = \rho_0$, the distance L_i to be used in the control is defined as shown in Eq. (8).

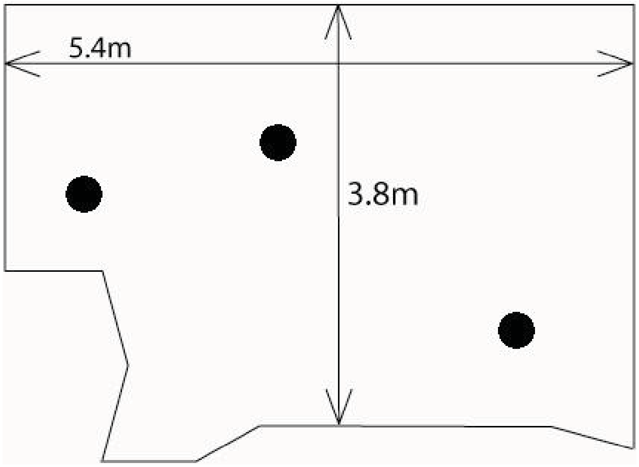


Figure 2. A REPRESENTATION OF THE REGION OF EXPLORATION. THREE ROBOTS ARE SHOWN TO SCALE.

$$L_i = \begin{cases} \ell_i & if\ \ \ell_i < L \\ L & if\ \ \ell_i \geq L \end{cases} \tag{8}$$

The virtual distance may then be used to define the controller for the robot, as shown in Eq. (9). For derivation of this controller from the potential function, see [8].

$$u_i = \begin{cases} \eta_t(\frac{1}{L_i} - \frac{L_{i-1}}{L_i^2}) + (v_{i+1} - 2v_i + v_{i-1}) & if\ L_i < L,\ L_{i-1} < L \\ \eta_t(\frac{1}{L} - \frac{L_{i-1}}{L^2}) + (v_{i-1} - v_i) & if\ L_i = L,\ L_{i-1} < L \\ \eta_t(\frac{1}{L_i} - \frac{L}{L_i^2}) + (v_{i+1} - v_i) & if\ L_i < L,\ L_{i-1} = L \\ 0 & if\ L_i = L,\ L_{i-1} = L \end{cases} \tag{9}$$

Testbed

The workspace for these experiments is confined to a planar region as shown in Fig. 2. The spills are simulated by covering some area of this region with a material with contrasting color. Several different configurations of spills are considered in this work, including convex and non-convex shapes, shapes with different areas, and contiguous or non-contiguous areas.

The agents in these experiments are iRobot Create® robots, which are differential-drive robots with wheelbase of 258*mm*. Control of these robots is accomplished via serial communication utilizing iRobot's "Open Interface" specification. The Open Interface specification allows access to all of the robot's sensor data. These robots have an external diameter of 35*cm*; in the current implementation, the calculation of avoidance potential functions does not take this into account, so the robots do occasionally collide (as indicated by their bump sensors). When a bump occurs (with an obstacle or a bump), the robots back up and turn a small amount.

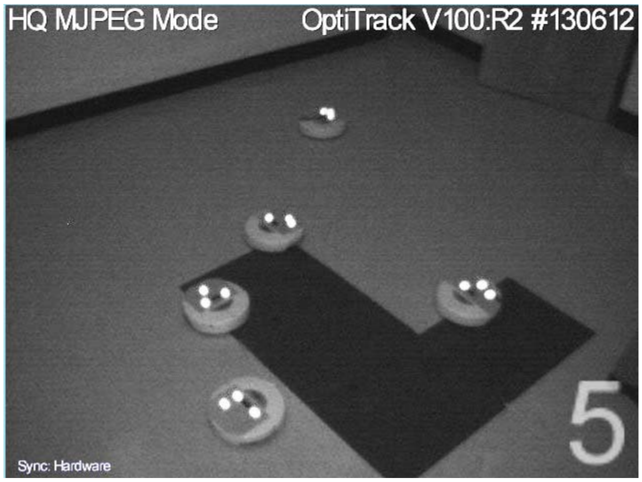


Figure 3. PERSPECTIVE FROM CAMERA #5 WITHIN THE SYSTEM.

The robots are wirelessly controlled via an off-board computer. The robots utilize the Bluetooth Adapter Module from Element Direct, made specifically for Create robots. In this architecture, the decision-making processes of all of the robots can be accomplished on the same physical computer, simplifying the development process. The processes remain distinct, however, and the control could equivalently be physically distributed to distinct, dedicated computers.

The spill sensors utilized are the Create's built-in down-looking IR proximity sensors, called "cliff sensors". The cliff sensors are used in the nominal operational functions provided on the robot as safety logic to detect stairs or other drop-offs to be avoided. For this experiment, the full-range sensor data will be utilized rather than the Boolean data, allowing calibration for different "colors" of spills.

A simulated sensor for each robot is provided by the arena's embedded motion capture system. With the ability to track each robot's position and orientation, the motion capture system yields GPS-like localization information as well as nearest-neighbor data. The motion capture system from NaturalPoint, called OptiTrack, is comprised of active-illumination infrared cameras and a software package, providing 6-DOF data for defined sets of markers within the field of view. Each robot is equipped with three reflective markers (each with a different pattern for identification; see [12, 13] for alternative solutions) to provide full 6-D measurement capability. A sample active-IR image of multiple robots in the field is shown in Fig. 3. The markers are clearly visible in this full-frame image, as are the spill and robot shapes. The robots in this experiment are constrained to a plane, thus the out-of-plane distance measurement and the roll and pitch orientation measurements are unnecessary (and should be unchanging).

Due to the lack of available sensors on the Create to sense neighbors, the motion capture system provides an external surrogate. Upon capturing a frame of data, the system computes all robot-to-robot distances. Based on an input parameter (ρ_c in later sections), the system applies a threshold to determine which

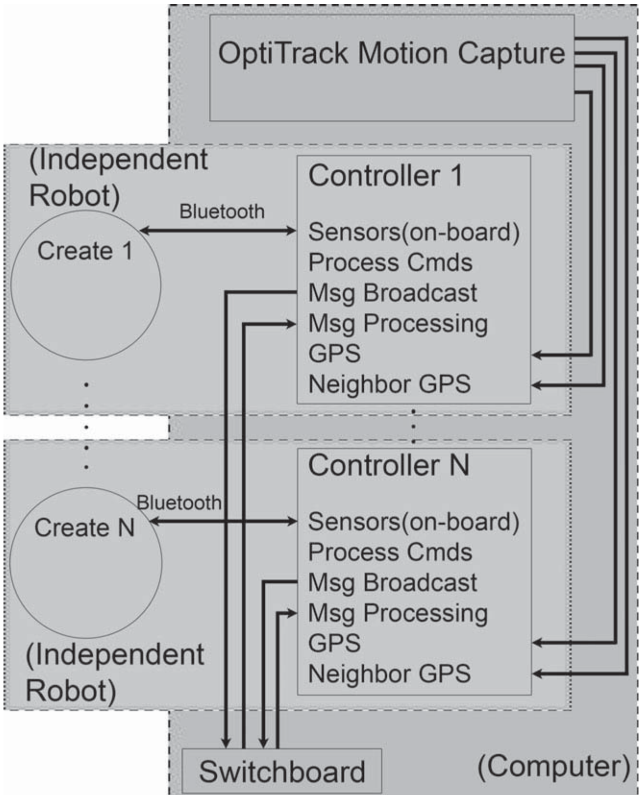


Figure 4. DIAGRAM OF PHYSICAL AND VIRTUAL BOUNDARIES IN THE MULTI-ROBOT CONTROL ARCHITECTURE.

robot can "hear" each other, e.g., those robots who are within communication range. The robots individually access their own connectivity data from the motion system, and can then "receive" messages from those robots. This communication mapping defines the robots' local graphs. The motion capture system additionally applies the user-specified ρ_r to determine which robots can "see" each other.

A diagram of virtual and physical boundaries is given in Fig. 4. Note that while all of the processing occurs on the same computer, the processes are completely independent of one another except for the simulated broadcast messages.

EXPERIMENTS

The experimental environment may be described as *mixed-simulation*, referring to the use of real hardware and real sensors combined with some simulated components. The simulated components are intended to allow greater configuration control and also reduce the financial burden and system integration and development time required.

Setup

Each experiment utilizes between three and five autonomous ground vehicles and one or two simulated spills. The "spills" in this experiment are formed from black material laid on the floor. Two different shapes are used, and other shapes are constructed by allowing spills to overlap. In Fig. 5, five Create robots and one non-convex spill are visible, alongside the "supervisor/switchboard" computer. Note that due to the sensor configuration on the Create robots, the "tracking" portion of the algorithm becomes an edge-following problem. Experiments with changing spill contours are conducted by manipulating one piece of material in an overlapped configuration.

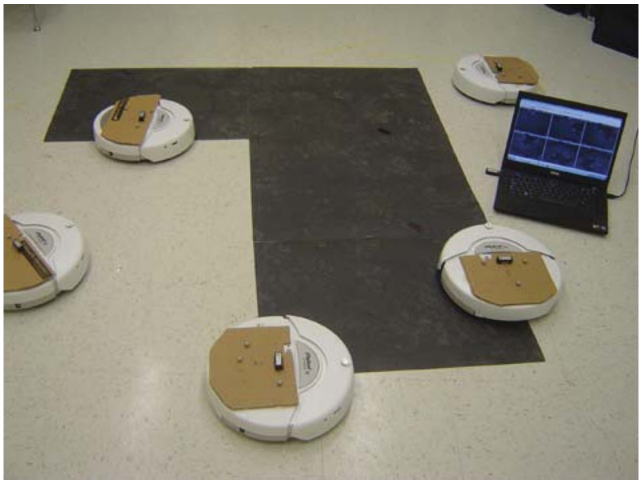


Figure 5. A WORKSPACE WITH FIVE IROBOT CREATES, ONE NON-CONVEX SPILL, AND THE CONTROL COMPUTER.

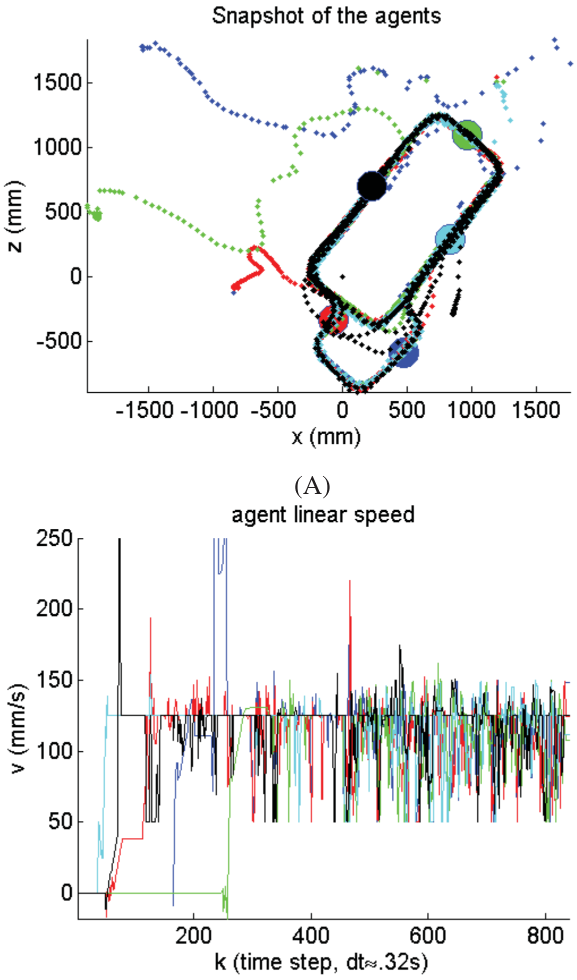
For the experiments reported here, the communication range is $\rho_c = 2m$, and the avoidance influential range ρ_0 and track-state influence range are $\rho_0 = L = 1m$.

Results

Results are presented from several experimental sets. The experiment shown in Fig. 6 utilizes five robots and a shrinking spill. Initially, the spill is configured in an L shape, similar to the shapes visible in Fig. 2 and Fig. 5. Once all robots are in Track state, the small leg of the spill is manipulated to condense the spill into a large rectangle. The perimeter of the large rectangle is roughly $5.6m$, larger than $N * L$, so the robots cannot achieve full equilibrium, although their velocities do converge.

The experiment shown in Fig. 7 utilizes three robots and only the small spill. The robots in this case quickly converge in Track state once the first robot finds the spill. Due to the geometry of the spill relative to the non-zero size of the robots and spurious communication dropouts (during which the robots stop for safety), the velocities do not converge asymptotically. However, the velocity variation is clearly centered about a nominal velocity, and the robot-robot spacing never deviates from equivalence significantly.

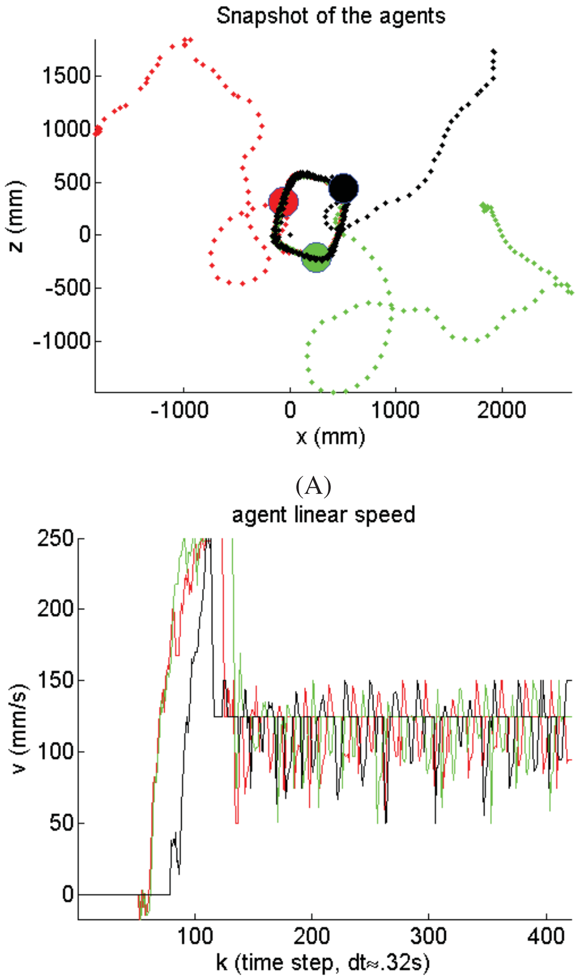
In these experiments, the communication is via Bluetooth, which becomes unreliable for four or more robots communicating with the switchboard. As a safety precaution, the robots stop driving when Bluetooth communication is (temporarily) lost;



(A)

(B)

Figure 6. EXPERIMENTAL RESULT FOR FIVE ROBOTS WITH A SHRINKING SPILL. (A) FULL TRAJECTORIES OF ALL ROBOTS, WITH SNAPSHOT LOCATIONS AT THE MIDPOINT OF THE RUN-TIME. (B) LINEAR SPEEDS OF ALL AGENTS.



(A)

(B)

Figure 7. EXPERIMENTAL RESULT FOR THREE ROBOTS WITH THE SMALLER SPILL. (B) FULL TRAJECTORIES OF ALL ROBOTS, WITH SNAPSHOT LOCATIONS AT THE MIDPOINT OF THE RUN-TIME. (B) LINEAR SPEEDS OF ALL AGENTS.

these occasional stops prevent the robots' velocities from converging asymptotically (as predicted by the algorithm). The velocities do, however, reach a stable, bounded value.

Continuing Work

The arena workspace is currently in transition to utilize direct, on-board control of the robots. Dell Mini10 netbook computers operate the Create robots and communicate with the central computer via wireless ethernet connections. Along with the on-board computers, the Create robots are now each equipped with a forward-looking Logitech webcam and a Hokuyo scanning laser rangefinder. The webcam provides a means for the Create to extend its spill-detection capability by employing tools from the OpenCV [14, 15] machine-vision software package. The laser rangefinder will be used to provide obstacle avoidance and direct detection of neighboring robots, while concurrently mapping the environment. Additional discussion of the multi-robot map-matching problem can be found in [16]).

Fig. 8 shows the latest configuration of the robots including camera, laser, and netbook. Preliminary experiments conducted with this new robot configuration are shown in Fig. 9 and Fig. 10. Note the improved convergence of velocities due to the decrease in data dropouts with the on-board netbooks. In Fig. 10, the velocities of some robots briefly increase (around k=700) when the smaller spill rejoins the larger spill and shrinks. The re-introduction of the fifth robot into the graph causes a disruption to the equilibrium, which quickly settles out. Note also that, in TRACK mode, the agents are subjected to a minimum speed (in this case, $V_{min} = 50mm/s$) to prevent the swarm from converging to a velocity that is too near to zero to be realized.

Parallel development is focused on the use of Gum-

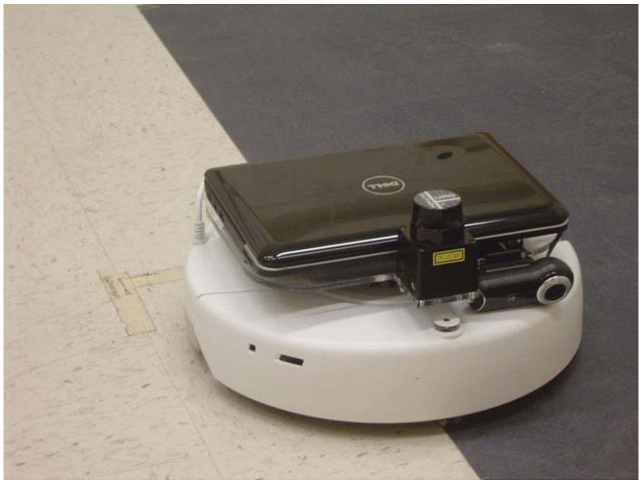


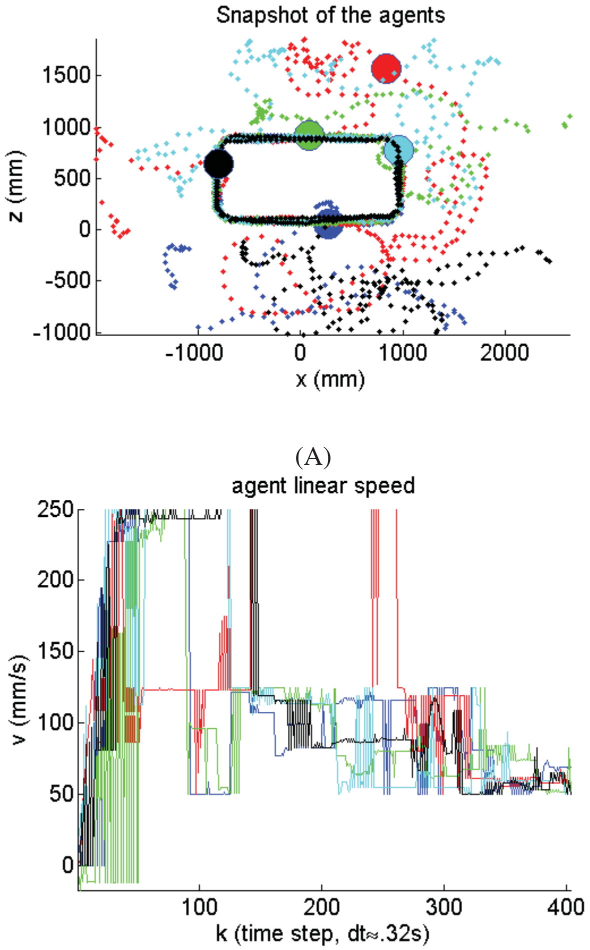
Figure 8. LATEST CONFIGURATION OF CREATE ROBOTIC AGENT, USING DELL MINI10 NETBOOK, LOGITECH QUICKCAM PRO 9000, AND HOKUYO URG-04LX-UG01 SCANNING LASER RANGEFINDER.

Stix Verdex small-form-factor computers controlling the Create robots via a Sticky Interface from Element Direct. The Gumstix computers have both Bluetooth and WiFi communication ability, allowing direct robot communication without switchboarding. This interface, along with a small USB hub powered off of the Create battery, likewise allows the use of the webcam and laser rangefinder sensors as described above.

This lab has recently acquired two Skybotix CoaX helicopters with Gumstix Overo computers on-board. The helicopters, equipped with USB cameras, will be integrated with this system to provide overhead observation of larger-scale areas, allowing extension of the ground robots' sensing ranges. Simulation, analysis, and planning of this system is underway for the appropriate position control of the helicopters to provide maximum utility to the system, drawing some inspiration from [17, 18].

To aid in the development of the aerial tracking capability, this lab is developing a more indoor-safe platform. Based on a radio-controlled blimp, this overhead system utilizes Gumstix Verdex and Robostix boards. The Verdex computer manages the control and communication with the system, while the Robostix interfaces with a lightweight webcam and PING ultrasonic range sensors for navigational control and altitude maintenance.

Use of the autonomous aerial vehicles as surveyors will yield direct, noisy measurements of perimeter length and number of spills, as well as coarse information about the ground robot positions. This information could be utilized to actively manage the number of agents participating in the surveillance of the perimeters; for example, some ground vehicles could be maintained in an "On-call" state, not expending energy until it is needed for perimeter coverage.



(A)

(B)

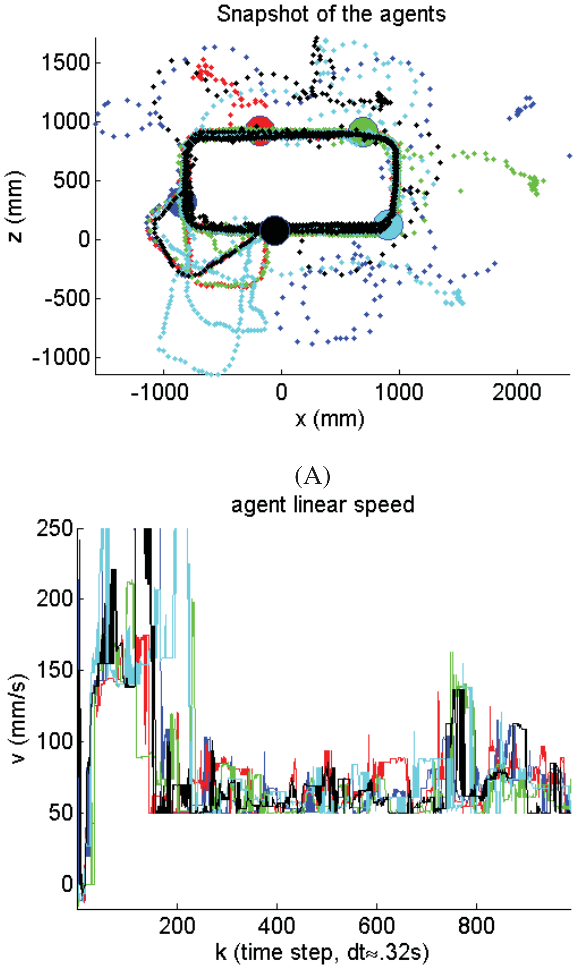
Figure 9. EXPERIMENTAL RESULT FOR FIVE ROBOTS WITH A STATIONARY LARGE SPILL. (A) FULL TRAJECTORIES OF ALL ROBOTS, WITH SNAPSHOT LOCATIONS AT THE MIDPOINT OF THE RUN-TIME. (B) LINEAR SPEEDS OF ALL AGENTS.

CONCLUSION

A fully distributable control methodology for detection and monitoring of dynamic perimeters has been developed and demonstrated. This control paradigm has been implemented in a mixed-simulation environment utilizing independent ground vehicles with limited communication. The practical issues related to such a system have been addressed, and work is continuing to develop a completely independent swarm that does not rely on the external motion-tracking system.

ACKNOWLEDGMENT

This work was supported by the Army Research Office under grant numbers W911NF-08-1-0106 and W911NF-09-0307.



(A)

(B)

Figure 10. EXPERIMENTAL RESULT FOR FIVE ROBOTS WITH A LARGE SPILL THAT GROWS, SPLITS INTO TWO SPILLS, THEN COLLAPSES TO ONE LARGE SPILL. (A) FULL TRAJECTORIES OF ALL ROBOTS, WITH SNAPSHOT LOCATIONS AT THE MIDPOINT OF THE RUN-TIME. (B) LINEAR SPEEDS OF ALL AGENTS.

REFERENCES

[1] Spears, W. M., Spears, D. F., Hamann, J. C., and Heil, R., 2004. “Distributed, physics-based control of swarms of vehicles”. *Autonomous Robots,* **17**, pp. 137–162.

[2] Hayes, A., Martinoli, A., and Goodman, R., 2001. “Swarm robotic odor localization”. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IRAS’01), Vol. 2, pp. 1073–1078.

[3] Casbeer, D. W., Kingston, D. B., Beard, R. W., and McLain, T. W., 2006. “Cooperative forest fire surveillance using a team of small unmanned air vehicles”. *International Journal of Systems Science,* **37**, pp. 351–360.

[4] Kingston, D., Beard, R., and Holt, R., 2008. “Decentralized perimeter surveillance using a team of UAVs”. *Robotics, IEEE Transactions on,* **24**(6), Dec., pp. 1394–1404.

[5] Clark, J., and Fierro, R., 2007. “Mobile robotic sensors for perimeter detection and tracking”. *ISA Transactions,* **46**(1), pp. 3 – 13.

[6] Cruz, D., McClintock, J., Perteet, B., Orqueda, O., Cao, Y., and Fierro, R., 2007. “Decentralized cooperative control - a multivehicle platform for research in networked embedded systems”. *Control Systems Magazine, IEEE,* **27**(3), June, pp. 58–78.

[7] Bruemmer, D., Dudenhoeffer, D., Anderson, M., and McKay, M., 2002. “A robotic swarm for spill finding and perimeter formation”. *Spectrum*.

[8] Zhang, G., Garg, D. P., and Fricke, G. K., 2010. “Hazardous spill perimeter detection and monitoring via multiple autonomous mobile robotic agents”. In Proceedings of the 3rd ASME Dynamic Systems and Control Conference (DSCC2010).

[9] Leonard, N., and Fiorelli, E., 2001. “Virtual leaders, artificial potentials and coordinated control of groups”. In Proceedings of the 40th IEEE Conference on Decision and Control, (CDC2001), Vol. 3, pp. 2968 –2973.

[10] Tanner, H., Jadbabaie, A., and Pappas, G., 2003. “Stable flocking of mobile agents, part I: fixed topology”. In Proceedings of the 42nd IEEE Conference on Decision and Control (CDC2003), Vol. 2, pp. 2010–2015.

[11] Tanner, H., Jadbabaie, A., and Pappas, G., 2003. “Stable flocking of mobile agents part II: dynamic topology”. In Proceedings of the 42nd IEEE Conference on Decision and Control (CDC2003), Vol. 2, pp. 2016–2021.

[12] Fricke, G. K., Milutinović, D., and Garg, D. P., 2009. “Sensing and estimation on an experimental testbed for swarm robotics”. In Proceedings of the 2nd ASME Dynamic Systems and Controls Conference (DSCC’09).

[13] Fricke, G. K., and Garg, D. P., 2010. “Discrimination and tracking of individual agents in a swarm of robots”. In Proceedings of the 2010 American Control Conference (ACC2010).

[14] Bradski, G., 2000. “The OpenCV Library”. *Dr. Dobb’s Journal of Software Tools*.

[15] Bradski, G., and Kaehler, A., 2008. *Learning OpenCV*. O’Reilly Media Inc.

[16] Fricke, G. K., Caccavale, A. W., and Garg, D. P., 2010. “Mobile sensor frame mapping via vision and laser scan matching”. In Proceedings of the International Symposium for Resilient Control Systems (ISRCS2010).

[17] Holland, O., Woods, J., De Nardi, R., and Clark, A., 2005. “Beyond swarm intelligence: the UltraSwarm”. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS2005), pp. 217–224.

[18] How, J., Bethke, B., Frank, A., Dale, D., and Vian, J., 2008. “Real-time indoor autonomous vehicle test environment”. *Control Systems Magazine, IEEE,* **28**(2), April, pp. 51–64.

A.2.10 A Sampling Approach to Modeling and Control of a Large-Size Robot Population

The following paper (on the next 7 pages) was co-authored by Dejan Lj. Milutinović and Devendra P. Garg, and appeared in the Proceedings of the 2010 ASME Dynamic Systems and Control Conference, Volume 2, on pages 631–637, published and presented in September 2010.

Proceedings of the ASME 2010 Dynamic Systems and Control Conference
DSCC2010
September 12-15, 2010, Cambridge, Massachusetts, USA

DSCC2010-4121

A SAMPLING APPROACH TO MODELING AND CONTROL OF A LARGE-SIZE ROBOT POPULATION

Dejan Milutinović [*]
Applied Mathematics and Statistics Department
University of California
Santa Cruz, California 95064
Email: dejan@soe.ucsc.edu

Devendra P. Garg
Mechanical Engineering and Materials Science Department
Duke University
Durham, North Carolina, 27708
dpgarg@pratt.duke.edu

ABSTRACT

Motivated by the close relation between estimation and control problems, we explore the possibility to utilize stochastic sampling for computing the optimal control for a large-size robot population. We assume that the individual robot state is composed of discrete and continuous components, while the population is controlled in a probability space. Utilizing a stochastic process, we can compute the state probability density function evolution, as well as use the stochastic process samples to evaluate the Hamiltonian defining the optimal control. The proposed method is illustrated by an example of centralized optimal control for a large-size robot population.

NOMENCLATURE

$\mathbb{R}, \mathbb{R}^n$ set of real numbers, set of real number vectors of dimension n.

Q set of discrete states, i.e., integer indexes $\{1,2,3,\dots\}$

U_{ad} set of admissible control.

$\rho(x,t)$ probability density function (PDF) of the hybrid state at time t. This variable is a vector of functions; it depends on $x \in X$ and $t \in \mathbb{R}$, but x and t are frequently omitted in expressions.

$\rho_i(x,t)$ the PDF component corresponding to the discrete state i, $i \in Q$.

$\pi(x,t)$ the adjoint state distribution.

$\bar{\pi}$ the discrete approximation of the adjoint state distribution.

$\phi(x,t)$ the adjoint state PDF.

$P_i(x,t)$ the probability of the discrete state i, $i \in Q$.

$P_i^{\pi}(x,t)$ the probability of the discrete adjoint state i, $i \in Q$.

F_t the transition rate matrix.

λ_{ij} the component of the transition rate matrix $[F_t]_{ij} = \lambda_{ij}$.

F_u the transition rate matrix that depends on control vector u, $u \in U_{ad}$.

F_∂ the component of the linear operator F corresponding to the vector fields f_i of discrete states $i \in Q$.

$H(\rho,u,t)$ the PDF, the control and time dependent Hamiltonian, ρ, u and t are frequently omitted.

u^* the optimal control

E_ρ the expected value with respect to the state PDF ρ.

INTRODUCTION

Solutions of multi-robot control problems may be of enormous complexity because of the operating environment uncertainties, or a large number of redundant states and robots. For many years it has been known that the optimal control and optimal estimation problems are closely related [1]. For example, the linear quadratic regulator (LQR) and the Kalman filter (KF) estimator can be derived in the same optimal control framework. Having this in mind, it is expected that estimation methods based on statistical sampling, and employed for solving complex estimation problems [2], can contribute to solving complex control problems for a single and, more importantly, multi-robot systems under the presence of uncertainty.

Along this idea, Kappen et. al. [3, 4] used stochastic differential equations to model individual agents. Based on this description, it is possible to relate the Hamilton-Jacobi-Bellman partial differential equation with samples of the stochastic process trajectories and use the samples to define the stochastic optimal control of multi-agent systems. In this framework, the state of individual robots is continuous. However, the state of real robots is generally described by a combination of contin-

[*] Address all correspondence to this author.

uous and discrete variables, i.e., by a hybrid state. Therefore, it is more natural to describe the robot behavior using a hybrid automaton [5]. The automaton describes the discrete and continuous variables change in time, which depends on events influencing the robot behavior. In the case of a large-size multi-robot population, it becomes highly complex to predict events from the robot local environment. Because of that, we model a large-size robot population considering the stochastic hybrid model and study how it can be controlled.

In this paper, we consider a problem in which the presence of a large-size robot population in a desired region of operating space is maximized. This problem is formulated in a hybrid system framework in [6]. Its solution, based on the minimum principle for partial differential equations, is presented in [7, 8], and it is solved numerically when the presence of the robots is maximized along one dimension (1D).

The Hamiltonian, which defines the optimal control, includes integral terms that depend on the solution of a system of partial differential equations (PDE). This system of PDEs is in general difficult to evaluate and the numerical evaluation of integrals is prone to errors. However, we recognize that the problem solution can be simplified and propose to use samples of the stochastic processes to evaluate the Hamiltonian components from the expected values of the adjoint state distribution.

The direction of the research we are pursuing is considerably different from the stochastic optimal control work presented in [9]. There, stochastic processes have been used only as an analytical tool to map the stochastic process to be controlled into the finite state space, in which the optimization is performed. The benefit of using a solution based on sampling, i.e, computational statistical methods, is that control problems in robotics could be solved faster. This possibility also depends on the ability to implement sampling and computations with samples into the processor computing the control.

MODELING AND CONTROL FRAMEWORK

In the modeling framework we consider, the state of an individual robot at time t is uniquely defined by the couple $(x(t), q(t))$, $x \in X$, $X \in \mathbb{R}^n$, $q \in Q, Q = \{1, 2, ...N\}$. While in the discrete state (mode) $k \in Q$, the continuous state of a robot obeys the differential equation $\dot{x} = f_k(x,t)$. We also assume that switching among the discrete states, say from the state $k \in Q$ to the state $j \in Q$, $(k \neq j)$, is described by stochastic transition rates λ_{kj}, and that $x(t)$ is a continuous function of time. In other words, the continuous state just before the discrete state transition $x(t^-)$ is equal to the state $x(t^+)$ after the state transition. This very general model of an individual robot is illustrated in Fig. 1 and the modeling framework we are applying here is detailed in [8].

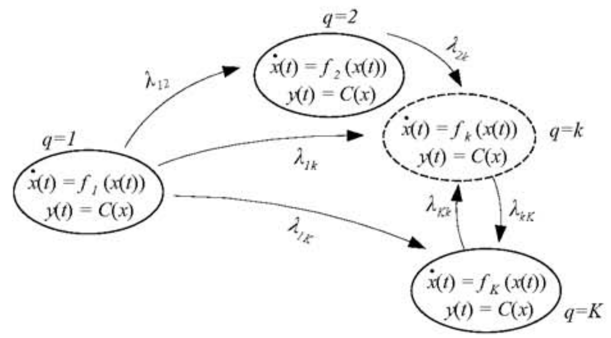


Figure 1. STOCHASTIC HYBRID AUTOMATON MODEL OF A ROBOT IN A PROBABILISTIC FRAMEWORK: DISCRETE STATE q; CONTINUOUS STATE x VECTOR FIELD f_k, $k \in Q$ DESCRIBES THE CHANGE OF THE CONTINUOUS STATE; STOCHASTIC TRANSITION RATES λ_{kj}, $k, j \in Q$ DESCRIBE THE MODE SWITCHING; y IS THE MEASURABLE OUTPUT; IF THE FULL CONTINUOUS STATE OF THE ROBOT IS MEASURABLE, C IS THE UNITY MATRIX.

Recognizing that the state of an individual robot is composed of discrete and continuous components, the state probability density function (PDF) is a vector of functions $\rho(x,t) = [\rho_1(x,t)\ \rho_2(x,t) \ldots \rho_N(x,t)]'$ [8]. Each component $\rho_i(x,t)$ corresponds to the discrete state i, and the symbol (') denotes the vector transpose. The state PDF satisfies

$$\sum_{i \in Q} \int_X \rho_i(x,t)dx = \sum_{i \in Q} P_i(t) = 1, \text{ where } P_i(t) = \int_X \rho_i(x,t)dx \tag{1}$$

where $P_i(t)$ is the probability of the discrete state i at the time point t. Let us define the vector of discrete probabilities $P(t) = [P_1(t), P_2(t), \ldots P_N(t)]'$, then evolution of the probability vector is given by:

$$\dot{P}(t) = F_t(t)P(t), \text{ where } [F_t]_{ij} = \lambda_{ij}(t) \tag{2}$$

with matrix F_t defining the transition rates among the discrete states. In general, the correspondence between the matrix F_t members $[F_t]_{ij}$ and the transition rates λ_{ij} is not one-to-one. Assuming that the transition rates depend on the vector $u(t) = [u_1(t)\ u_2(t)\ \ldots u_M(t)]'$ of variables $u_i, i = 1, 2 \ldots M$, we can define the transition rate matrix as a function of the vector $u(t)$, i.e., $F_t(t) = F_u(u(t))$. Consequently, the vector of the discrete state probabilities obeys [10]:

$$\dot{P}(t) = F_u(u)P(t) \tag{3}$$

Moreover, it can be proven [8] that the state PDF obeys the following system of partial differential equations (PDE):

$$\frac{\partial \rho(x,t)}{\partial t} = F(u)\rho(x,t) = (F_u(u(t)) + F_\partial)\rho(x,t) \tag{4}$$

where F_∂ is the diagonal linear differential operator. When the operator F_∂ is applied to $\rho(x,t)$, it results in:

$$[F_\partial\rho(x,t)]_{ij} = \begin{cases} -\nabla\cdot(f_i\rho_i(x,t)), i = j \\ 0, i \neq j \end{cases} \quad i,j = 1,2\ldots N \tag{5}$$

Taking into account that the state PDF evolution $\rho(x,t)$ depends on the vector $u(t)$, we can formulate the optimal control problem in the probability space using the cost function:

$$J = \int_X w'(x)\rho(x,T)dx \tag{6}$$

In this respect, the optimal control problem is the optimization problem:

$$u^*(t) = \max_{u(t)\in U_{ad}} J = \max_{u(t)\in U_{ad}} \int_X w(x)\rho(x,T)dx \tag{7}$$

Alternatively, to avoid the singular control problems [8], we can consider the optimal control that includes the term penalizing the control:

$$u^*(t) = \max_{u(t)\in U_{ad}} J = \max_{u(t)\in U_{ad}} \int_X w(x)\rho(x,T)dx + \varepsilon\int_0^T u'(t)u'(t) \tag{8}$$

Anyway, the solution of this problem is a sequence of the optimal control $u^*(t)$, from the set of admissible control U_{ad}, such that the cost function is maximized. By a suitable choice of the weighting function $w(x)$, the cost function can be used to find the optimal control maximizing probability of the robot presence in the desired region of the robots' operating space.

The optimal control maximizing the criterion (6) is a special case of a more general optimal control problem of the evolution equation [11]. Under the condition that the operator $F(u)$ is bounded, i.e., $\|F(u(t))\| < \infty$, the minimum principle for PDEs can be applied [11]. According to the minimum principle, the optimal control $u^*(t)$ satisfies:

$$u^*(t) = \arg\min_{u\in U_{ad}} H(\rho(x,t),u(t),t) \tag{9}$$

In other words, for the optimal state PDF trajectory $\rho^*(x,t)$, the optimal control minimizes the Hamiltonian at each time point. The Hamiltonian is:

$$H(\rho(x,t),u,t) = \langle\pi(x,t),F(u)\rho(x,t)\rangle \tag{10}$$

where brackets $\langle\cdot,\cdot\rangle$ denote the scalar product of function vectors defined as:

$$\langle p(x),q(x)\rangle = \int_X p'(x)q(x)dx = \int_X \sum_i p_i(x)q_i(x)dx \tag{11}$$

The function vector $\pi(x,t)$ is the so-called adjoint state distribution and satisfies:

$$\frac{\partial\pi(x,t)}{\partial t} = -F'(u)\pi(x,t) \tag{12}$$

$$\pi(x,T) = -w(x) \tag{13}$$

The major difficulty in computing the optimal control is in evaluation of integrals (11) and corresponding PDE system solutions (4) and (12). Based on the definition of the scalar product (11), the Hamiltonian can be expressed as:

$$H(\rho(x,t),u,t) = \langle\rho(x,t),F'(u)\pi(x,t)\rangle \tag{14}$$

,i.e., $$H(\rho(x,t),u,t) = \sum_i \int_X \rho_i(x,t)[F'(u)\pi(x,t)]_i dx \tag{15}$$

where $[\cdot]_i$ denotes the ith row of the vector. In the following section, we will explain how the evolution of the state PDF ρ, as well as expression (15) can be computed using the stochastic sampling propagator.

STOCHASTIC SAMPLING PROPAGATOR

The evolution of the large-size population probability density function $\rho(x,t)$ is described by the PDE system (see Eq.4). One way to obtain the evolution $\rho(x,t)$ is to solve the PDE system forward in time starting from the initial condition $\rho(x,0) = \rho^0(x)$. We propose an approach to computing the evolution $\rho(x,t)$ based on stochastic trajectories of the hybrid state (x,q) evolution resulting from the model presented in Fig. 1. To account for the fact that the transition rates can change in time, we assume that the control is a piecewise constant function of time discretized with the sample time ΔT. The basis for the proposed algorithm is the Gillespie's stochastic simulation algorithm [12].

To generate the trajectory of (x,q), we need to generate the initial state $(x(0),q(0))$ from the state PDF $\rho(x,0) = \rho^0(x)$. Probability $P_i(t)$ of $q(t) = i$ is:

$$P_i(t) = \int_X \rho_i(x,t)dx \tag{16}$$

Therefore, the random variable $q(0) = i$ should be generated from the discrete state probability distribution represented by the vector of discrete state probabilities $P(0) = [P_1(0)\ P_2(0)\ \ldots P_N(0)]'$. Symbolically, we will represent it as:

$$q(0) = i \sim P(0) \tag{17}$$

Once the initial discrete $q(0)$ state is defined, the continuous variable $x(0)$ can be generated from the corresponding $\rho_i(x,0)$ component of the state PDF ,i.e., from the probability $\mathcal{P}$ of $x(t)$ given that $q(t)=i$ and $t=0$:

$$x(0) \sim \mathcal{P}\{x|q(t)=i, t=0\} = \rho_i(x,0)/P_i(0) \tag{18}$$

Whenever the discrete state is $q(t)=i$, the evolution of the continuous state x obeys $\dot{x}=f_i(x)$. Therefore, generating trajectory $(x(t),q(t))$ reduces to the problem of generating the state transitions of the discrete state $q(t)$. Let us assume that at time $t=t_s$, $t_s \in [(k-1)\Delta T, k\Delta T)$, the hybrid state is $(x(t),q(t))$; then, the time instant at which the state changes t_c can be generated based on the following two rules:

(a) $t_c = t_s + t_t$, $t_t \sim e^{-t\sum_j \lambda_{ij}(k-1)}$, under the condition that $t_c < k\Delta T$. If the condition is not satisfied, apply rule (b).
(b) $t_c = k\Delta T + t_t$, $t_t \sim e^{-t\sum_j \lambda_{ij}(k)}$, under the condition that $t_c < (k+1)\Delta T$. If the condition is not satisfied, increase k by 1. Apply rule (b) until the condition is satisfied.

These two rules define the time point t_c at which the jump from the discrete state i to the discrete state j happens, but do not specify the variable j. The state j needs to be sampled from the discrete state probability density function, i.e., from the probability $\mathcal{P}$ of $q(t^+)=j$, given that $q(t)=i$ provided in the vector of the discrete probability distribution with $N-1$ elements:

$$j \sim \mathcal{P}\{q(t)|q(t^-)=i\} = \underbrace{\left[\frac{\lambda_{i1}}{\sum_{n=1}^{N}\lambda_{in}}, \frac{\lambda_{i2}}{\sum_{n=1}^{N}, \lambda_{in}} \cdots \frac{\lambda_{iN}}{\sum_{n=1}^{N}\lambda_{in}}\right]}_{N-1} \tag{19}$$

The above algorithm can be used to generate a single trajectory for the stochastic model shown in Fig. 1. In the limit of a large number of samples, the normalized density of trajectory points will correspond to the solution of the PDE system given by Eq. 4. In this respect, the stochastic simulation is a computational propagator of the evolution $\rho(x,t)$ and we can denote it as:

$$\frac{\partial \rho}{\partial t} = F_{sim}(u(t))\rho \tag{20}$$

HAMILTONIAN EVALUATION

Let us assume that the total number of the trajectories we use to propagate the state PDF ρ is N_{samp}, that $x_k(t)$ and $q_k(t)$ denote the continuous state and discrete states of trajectory k at time t, $k=1\ldots N_{samp}$. At a given time point t, among N_{samp} trajectories only $N_i(t)$ trajectories are in the discrete state i, and naturally, $\sum_i N_i(t) = N_{samp}$, $\forall t$. From the state PDF normalizing condition (1), we can conclude that:

$$\frac{1}{P_i(t)}\int_X \rho_i(x,t) = 1 \tag{21}$$

Following, the expected value of $[F'(u)\pi(x,t)]_i$ under the condition that the discrete state $q(t)=i$ is

$$E\{F'(u)\pi(x,t)|q(t)=i\} = \frac{1}{P_i(t)}\int_X \rho_i(x,t)[F'(u)\pi(x,t)]_i dx \tag{22}$$

and can be approximated as:

$$E\{F'(u)\pi(x,t)|q(t)=i\} \approx \frac{1}{N_i(t)}\sum_k [F'(u)\pi(x_k,t)]_i \delta(q_k(t)-i) \tag{23}$$

where $\delta(q_k(t)-i)=1$ if $q_k(t)=i$, and zero elsewhere. Conseqently, the Hamiltonian (15) can be expressed as:

$$\begin{aligned} H\ (\rho(x,t),u,t) &= \textstyle\sum_i P_i(t) E\{F'(u)\pi(x,t)|q(t)=i\} \\ &\approx \textstyle\sum_i P_i(t)\frac{1}{N_i(t)}\sum_k [F'(u)\pi(x_k,t)]_i \delta(q_k(t)-i) \end{aligned} \tag{24}$$

In the limit of a large number of samples $P_i(t) \approx N_i(t)/N_{samp}$; therefore, the Hamiltonian value can be estimated using the following expression:

$$\hat{H}(\rho(x,t),u,t) = \frac{1}{N_{samp}}\sum_i\sum_k [F'(u)\pi(x_k,t)]_i \delta(q_k(t)-i) \tag{25}$$

where $\sum_k$ denotes the sum over all trajectories and $\sum_i$ over all discrete states. This expression is exact in the limit of a large number of samples N_{samp}. To illustrate and verify the algorithm for generating stochastic trajectories $(x(t),q(t))$ and computing the Hamiltonian components, we use a 1D example in the next section.

1D EXAMPLE

The stochastic model presented in Fig. 2b illustrates the state PDF evolution of a large-size robot population along one dimension (Fig. 2a), in which u_1, u_2 and u_3 correspond to stochastic rates of the commands: move-left, move-right and stop. In this example, $k_1=-0.5$ and $k_2=0.25$. The control $u(t)=[u_1(t)\ u_2(t)\ u_3(t)]$ is computed as the optimal control based on the minimum principle and Hamiltonian presented in the previous section.

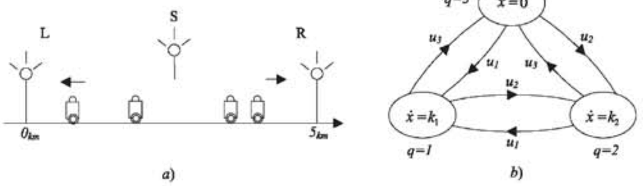


Figure 2. 1D EXAMPLE [7, 8]

The cost function is:

$$J = \int_X w'(x)\rho(x,t)dx + \varepsilon \int_0^T u_1^2(t) + u_2^2(t) + u_3^2(t)dt \qquad (26)$$

where $\varepsilon = 10^{-7}$, the weighting $w(x) = [0\,0\,w_3(x)]'$ and the initial condition $\rho(x,0) = [0\ \ 0\ \rho_3(x,0)]'$ are defined by:

$$w_3(x) = \begin{cases} \frac{1}{\sqrt{0.01}} exp(-\frac{(x-1.75)^2}{0.01}),\ 1.25 < x < 2.25 \\ 0,\ otherwise \end{cases} \qquad (27)$$

$$\rho_3(x,0) = \begin{cases} \frac{1}{\sqrt{0.02\pi}} exp(-\frac{(x-2.5)^2}{0.02}),\ 2 < x < 3 \\ 0,\ otherwise \end{cases} \qquad (28)$$

The optimal control sequence $u^*(t) = [u_1^*(t)\ u_2^*(t)\ u_3^*(t)]$ in the time interval $0 < t < 3$ is defined by:

$$u_1^*(t) = \begin{cases} 2,\ 0.21 < t < 1.74 \\ 0, elsewhere \end{cases},\ \ u_2^*(t) = 0 \qquad (29)$$

$$u_3^*(t) = \begin{cases} 2,\ 1.71 < t < 3 \\ 0, elsewhere \end{cases} \qquad (30)$$

The evolution of the state PDF for this system under the control $u^*(t)$ is presented in Fig. 3. We present only $\rho_1(x,t)$ and $\rho_3(x,t)$ because under this control $\rho_2(x,t) = 0,\ \forall t$.

For the illustration, we generated 10 stochastic trajectories of the continuous variable x (see Fig. 5) under the control $u^*(t)$. The evolution of the discrete state q can be observed from the trend in x. When x decreases, the discrete state is 1, and when it remains constant, the state is $q = 3$. It is worth mentioning that, among these 10 trajectories, there is one for which $x(t)$ is constant. The small pick around the point 2.5 in the right panel of the Fig.3 at $t = 3$, confirms that the probability of such trajectories is non-zero, but it is small.

To obtain the state PDF $\rho(x,t)$, i.e., its components $\rho_i(x,t)$ at a specific time point t, we need to collect points $x(t)$ and estimate components $\rho_i(x,t)$. It is obvious that 10 trajectories cannot provide a good estimate of $\rho(x,t)$ For this reason, we generated 10^5 trajectories and computed the histogram probability density function estimate. That means that we discretized the x axis into intervals of the length Δx and counted how many points fell into a specific region. Finally, we normalized the histogram so that the estimated $\rho(x,t)$ is normalized to 1. Our results are presented in Fig. 4. As expected, the match between the numerical PDE system solution and the result obtained from stochastic trajectories is exact. There are only negligible discrepancies due to data sampling from a finite number of trajectories.

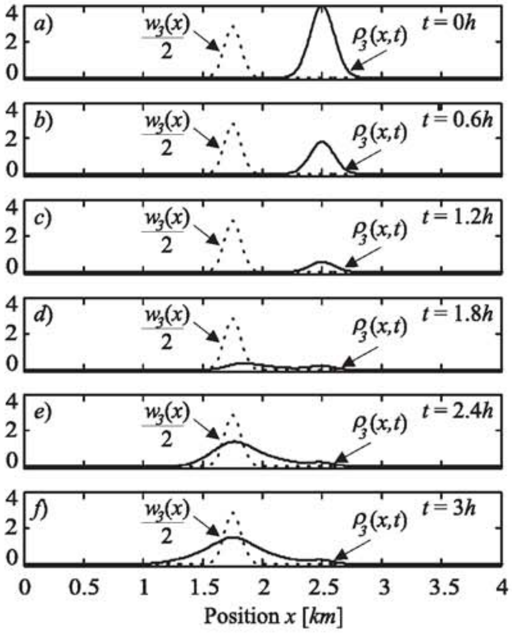


Figure 3. THE FINITE ELEMENT SOLUTION OF THE STATE PDF EVOLUTION FOR THE 1D EXAMPLE UNDER THE OPTIMAL CONTROL $u^*(t)$, 500 POINTS [7,8]

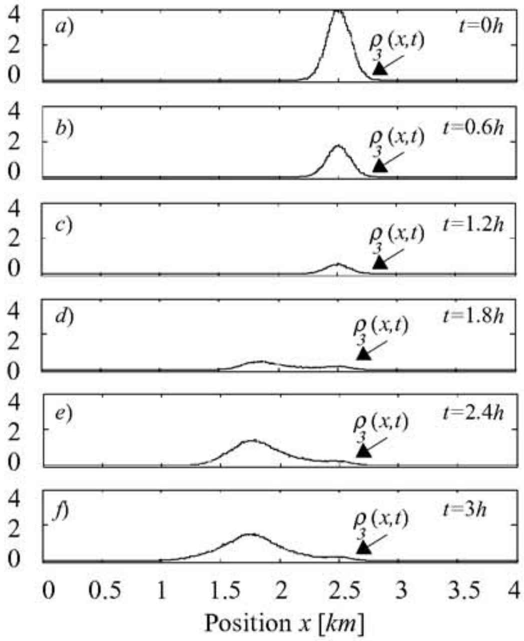


Figure 4. THE STOCHASTIC SIMULATION SOLUTION OF THE STATE PDF EVOLUTION FOR THE 1D EXAMPLE UNDER THE OPTIMAL CONTROL $u^*(t)$, 10^5 SAMPLES

In Table 1, we provide the time our MATLAB code takes to compute the state PDF evolution based on the stochastic simulation approach (Fig. 4) and the time it takes to compute the evo-

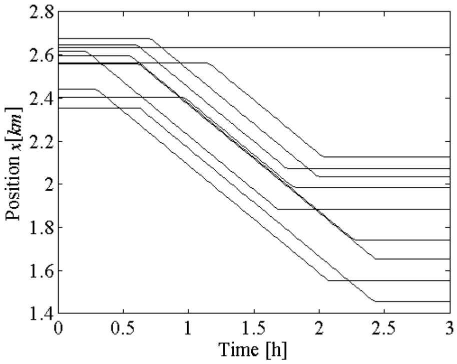


Figure 5. A RANDOM SET OF 10 TRAJECTORIES RESULTING FROM THE STOCHASTIC SIMULATION UNDER THE OPTIMAL CONTROL $u^*(t)$

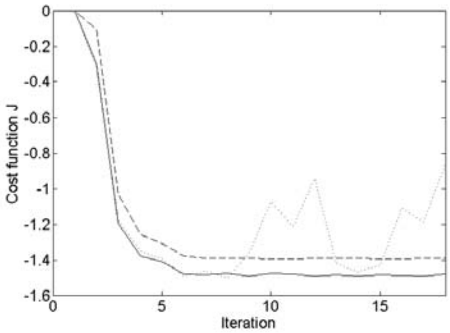


Figure 6. COST FUNCTION J ITERATIONS COMPUTED WITH THE ALGORITHM [8] WITH THE HAMILTONIAN EVALUATED BASED ON THE PDE SOLUTIONS (DASHED), AND ESTIMATED BASED ON 10^4 (DOTTED) AND 10^5 (SOLID) STOCHASTIC SAMPLES.

lution based on the finite element (FE) approach with 500 points (Fig. 3). We can see that the time for the stochastic simulation approach increases roughly linearly with the number of samples and that the time of the FE approach can be reached if we use approximately $3 \cdot 10^5$ samples.

For the purpose of computing the optimal control, the number of the samples we need depends on the convergence of the optimization algorithm to the solutions. Therefore, as the final test of the stochastic sampling propagator, we computed the optimal control based on the algorithm presented in [8], but using the stochastic samples for the Hamiltonian evaluation (25) instead of the Hamiltonian which is completely based on PDE systems solutions [8]. We can see from Fig. 6 that for 10^4 samples, the Hamiltonian (25) fluctuations are at such a level that the optimization algorithm does not converge, and we see it as large fluctuations of the cost function. While the fluctuations are intrinsic property of the stochastic samples evaluation, they can be smaller if we use a larger number of samples.

Table 1. TIME COMPARISON BETWEEN THE STOCHASTIC SIMULATION APPROACH AND THE FINITE ELEMENT (FE) SOLUTION WITH 500 POINTS

Number of samples	10^3	10^4	10^5	FE
Time (s)	0.48s	4.8s	44.8s	1363s

When we use 10^5 samples, the fluctuations are much smaller and the optimization algorithm converges. It is interesting to notice that the stochastic sampling evaluations result into the cost function value which is smaller that the value resulting from the PDE system solutions (see Fig. 6). This is because of discrete approximations involved into solutions of PDE systems, as well as approximation of integrals contributing to the cost function.

The average time that our MATLAB code takes for a single iteration based on the finite element method (500 points) is 1629s (Fig. 6, dashed line). Using the same optimization code, a single iteration in the stochastic-based method with 10^5 samples is 350s (Fig. 6, solid line). Under the conditions presented above, the stochastic-based method is in average 4.5x faster than its deterministic counterpart.

CONCLUSION

In this paper we considered a large-size robot population control problem that had been previously formulated and solved in a probability space utilizing systems of PDEs. Solving these PDEs is computationally expensive; therefore, having in mind that the PDEs are in close connection with the stochastic process to be controlled, we explore an opportunity to utilize the stochastic process samples to compute the control.

Our paper describes an algorithm for generating the stochastic process that can be used to propagate the state PDF of the robot population. We show that the algorithm predicts exactly the state PDF evolution and we derive expression for the Hamiltonian evaluation which involves the stochastic process samples. The Hamiltonian evaluated in this way can be used in iterations computing the optimal control as if it was computed based on the PDE system solutions. We also notice that the cost function resulting from utilizing stochastic processes has smaller values than the cost function computed based on PDEs. This means that the evaluations involving the stochastic process samples are closer to the true values.

In summary, we can conclude that utilizing stochastic processes for computing control of multi-robot systems considering discrete, as well as continuous robot states is possible. By embedding stochastic process generators into analog circuits and utilizing them in dedicated processors for computing control, complex stochastic optimal control problems can be solved efficiently and potentially exploited for real-time multi-robot systems control.

ACKNOWLEDGMENT

This research was supported by the NASA UARC Aligned Research Program (ARP) grant NAS2-03144 Task TO.084.0.MD.D and by the Army Research Office (ARO) grant number W911NF-08-0106.

REFERENCES

[1] Ho, Y., and Bryson, A. E., 1975. *Applied Optimal Control: Optimization Estimation and Control*. John Wiley & Sons.

[2] Robert, C., and Casella, G., 2004. *Monte Carlo Statistical Methods*. Springer.

[3] Kappen, H. J., 2005. "Linear theory for control of nonlinear stochastic systems". *Physical Review Letters,* **95**(20), p. 200201.

[4] van den Broek, B., Wiegerinck, W., and Kappen, B., 2008. "Graphical model inference in optimal control of stochastic multi-agent systems". *Journal of Artificial Intelligence Research,* **32**, pp. 95–122.

[5] der Schaft, A. V., and Schumacher, J., 2000. *An Introduction to Hybrid Dynamical Systems*. Springer-Verlag.

[6] Milutinović, D., Lima, P., and Athans, M., 2003. "Biologically inspired stochastic hybrid control of multi-robot systems". In Proceedings of the 11th International Conference on Advanced Robotics.

[7] Milutinović, D., and Lima, P., 2006. "Modeling and optimal centralized control of a large-size robotic population". *IEEE Transactions on Robotics,* **22**, pp. 1280–1285.

[8] Milutinović, D., and Lima, P., 2007. *Cells and Robots : Modeling and Control of Large-Size Agent Populations*. Springer.

[9] Kushner, H. J., 1967. *Stochastic Stability and Control*. Academic Press, New York.

[10] Cassandras, C., and Lafortune, S., 1999. *Introduction to Discrete Event Systems*. Kluwer Academic Publ.

[11] Fattorini, H., 1999. *Infinite Dimensional Optimization and Control Theory*. Cambridge University Press.

[12] Gillespie, D., 1977. "Exact stochastic simulation of coupled chemical reactions". *Journal of Physical Chemistry,* **81**(25), pp. 2340–2361.

A.2.11 Modeling, Simulation, and Characterization of Distributed Multi-Agent Systems

The following paper (on the next 6 pages) was co-authored by Reed F. Young and Devendra P. Garg, and appeared in the Proceedings of the International Multi-Conference on Complexity, Informatics and Cybernetics, on pages 179–184, published and presented in March 2011.

MODELING, SIMULATION, AND CHARACTERIZATION OF DISTRIBUTED MULTI-AGENT SYSTEMS

Reed F. YOUNG

and

Devendra P. GARG

Mechanical Engineering & Materials Science Department
Duke University
Durham, North Carolina, 27708 USA
Email: {reed.young, dpgarg}@duke.edu

ABSTRACT

This paper explores the characterization of a multi-agent system through novel application of a potential force function that includes tuning coefficients. System parameters are varied against a pre-defined area mapping and target identification mission and evaluated against system performance metrics to isolate agent capability combinations most important to pre-mission planning functionality. Excursions were run in simulation iterations producing rich data sets for evaluation and reduction. Additional future research will investigate the impact of potential function tuning.

Keywords: multi-agent system, robot cooperation, mission planning, potential force functions

INTRODUCTION

The concept of organizing groups of robots into "multi-agent systems" (MAS) has existed for several years. Research in the field has centered on understanding the root mechanisms and control features that can define and drive the behavior and capability of the MAS architectures in terms such as task allocation, flocking or foraging. The study of analogous biological systems has also provided valuable insight. However, to date, there has been little work done with respect to understanding the capability of the MAS as a function of its composition given the option of pre-mission selection of the performance and capability of individual nodes, as well as opportunity to include extra-system information sources or sensors.

The ability of a MAS to perform complex missions beyond the capability of today's existing unmanned systems will be principally a function of three advancements. First is the reduction of the electronic and mechanical components of the physical configuration while increasing or maintaining desirable performance characteristics. The second is the characterization of the behavioral control mechanisms on the MAS, such that complex tasks can be performed with significant mission accomplishment in a dynamic set of boundary conditions, core capabilities, and constraints. Third, and the principal motivation for this research, is increasing the capability of the MAS itself as a function of composition. Due to the inherent flexibility and variety that robotic systems typically offer, it is merely intuitive that the strength of a MAS' capability will be in its diversity. The major benefit arising out of this research effort is to gain a better understanding of the impact that variations in the selected composition of the MAS can have on its resulting capabilities and performance with a focus on the composition-control scheme relationships.

BACKGROUND

From the inception of the development of robotic technologies, researchers have generally concluded that, with many if not most applications, there are significant advantages to be gained through the coordination and orchestration of multiple robotic entities [1, 2]. This is not difficult at all to imagine by simply observing Darwinian development in biological systems whereby two or four legs proved superior to one for locomotion; or separate arms each with a hand, fingers, and an opposing thumb proved superior for grasping and cooperatively manipulating objects. Further, it similarly is understandable that a small group of separate robotic entities (e.g., a small number of cooperating or coordinating unmanned ground vehicles) could also exhibit significantly desirable characteristics in certain circumstances. Five significant motivations for using multi-robot systems include: task complexity; task distribution; resource distribution; parallel processing; and robustness through redundancy [3]. Some researchers have explored the advantages gained for specific tasks such as exploration where now the task can be accomplished by numbers of agents surveying the maneuver space versus a single robot tasked with exploring the entirety by itself [4]. Indeed, these become valuable against overarching task accomplishment; however, not without burden. Multi-robot systems add communications overhead and workload with the requirement for inter-robot cooperation and knowledge. They also add complexity to path planning and collision avoidance schemes since now each robot's cooperating peer also exists as an obstacle to avoid [5]. Multi-robot systems, say, with on the order of 10 nodes, increase this complexity. Beyond that number, say tens or hundreds of nodes comprising "swarms," might illustrate orders of magnitude complexity increase.

Attempts [6-8] at defining MAS composition explored heterogeneous versus homogeneous systems, highlighting concepts such as hierarchic social entropy, diversity, and cooperative localization. Heterogeneous MAS have been used to explore unknown environments for the purpose of mapping and exploration [9]. Here, each MAS featured varying types of single-mode sensor payloads, including for example an infrared camera or sonar, coordinated in an occupancy grid Bayesian mapping algorithm. But again, the composition of the architecture was held constant.

The underlying control architecture plays an absolutely critical role in the efficiencies observed in the multi-agent versus single robot architectures [10-13]. Clearly, beyond the inherent capability of the electro-mechanical configuration as well as the sensor payloads, the single greatest contributor to mission efficiencies and task capabilities will exist in the capability and comprehensiveness of the control system. A wide variety of algorithms and schemes have been explored including feedback laws, leader follower, task-oriented mission planning, task reallocation and reordering, and deliberative and reactive behavior modeling.

AGENT KINEMATICS

Differential-wheeled robots (Figure 1) comprise a large segment of the experimental robot type used in research today. Quite simply, there are two drive wheels situated parallel and opposite to each other that can be independently controlled in terms of applied torque to dictate the robot's motion. The kinematic equations governing differential-wheeled robots are:

$$\dot{x}_i = v_i \sin(\phi_i) \tag{1}$$
$$\dot{z}_i = v_i \cos(\phi_i) \tag{2}$$
$$\dot{\phi}_i = \omega_i \tag{3}$$

where,

v_i is the linear velocity,

ω_i is the rotational velocity, and

x_i, z_i, ϕ_i are the absolute position and orientation, respectively.

Further, the position of the i-th robot can be represented by the vector:

$$r_i = [x_i, z_i]^T, r_i \in \boldsymbol{R}^2 \tag{4}$$

And the vectors representing N robots comprising the multi-agent system (MAS) is:

$$r_{MAS} = [r_1^T, r_2^T, \ldots, r_N^T]^T, r_{MAS} \in \boldsymbol{R}^2 \tag{5}$$

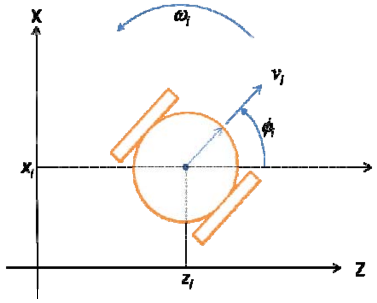


FIGURE 1 – Differential-wheeled Robot Kinematics

ROBOT CONTROL STRATEGY

Control via potential-field theory has proven to be a valuable basis upon which more complex control feedback influence can be implemented. In particular, potential-field theory affords a comprehensive and simple method to add factors that can be scaled either as independent input functions or dependently upon selected system features.

The basis of the potential field is to represent system control influences as independent or dependent forces [14]. Each force is calculated based on physical parameters (such as real distances) or on representative values (such as arbitrary repulsion to facilitate obstacle avoidance or attraction to a desired target location). In conjunction with the potential field, it is convenient to represent the operational space as an occupancy map, i.e., a two- or three-dimensional grid that sub-divides the space into identifiable cells. Each cell would then have characteristics such as a potential related to its occupancy status, or status as a target destination. Since potential is a function of relative proximity, the relative location of the cells would also be cataloged.

The goal of the control function for MAS will be to calculate the agent-unique potential force witnessed by each of the agents comprising the MAS. Force will define the resultant heading and speed that the individual agent would follow. The potential force existent on the Mth agent is given by:

$$\hat{F}_M = Q_M \left(\sum_{i=1}^{N} \frac{\mu_1 q_i}{\|r_i\|^2} \hat{r_i} + \sum_{i=1}^{M-1} \frac{\mu_2 Q_i}{\|r_i\|^2} \hat{r_i} \right) + \sum_{i=1}^{L} \mu_3 P_i \qquad (6)$$

where,

Q_M = inherent charge of the Mth agent,

Q_i = inherent charge of the ith agent,

q_i = inherent charge of the ith cell,

r_i = radius vector from the Mth agent to the ith agent or ith cell,

N = number of cells in the occupancy map,

M = total number of agents,

P = calculated force contribution from specified payloads, and

μ_x = tuning coefficients.

Then, the resultant heading angle of the Mth agent is given by:

$$\theta_M = \tan^{-1}\left(\frac{F_{Z\,M}}{F_{X\,M}}\right) \qquad (7)$$

The calculation of P will vary with payload type as the value and weighting of each payload type can and should have varying degrees of influence on the resultant potential force calculation.

Tuning coefficients (μ_x) provide a mechanism whereby the relative contribution of each of the potential forces can be directly tuned either as an independent variable, or dynamically tuned based on a feedback mechanism. This feature is particularly important with multi-agent systems due to scalability. Since the potential forces contributed by elements within the system are cumulative, it becomes very important to gain an awareness of the proportional contributions and subsequently be able to control them proactively. To illustrate, let us assume an occupancy grid of 120X240 cells or a total of 28,800 total cells. Each cell might offer a contributing charge value from 0 to 1 meaning a span of 28,800 charge units. If other contributors merely summed to a maximum of 1,000 charge units, the cells' contribution would easily overwhelm any potential contribution calculation.

The occupancy map cell characteristic vector represents the probability that the cell is occupied. A value of 1.0 indicates that there is a 100 percent chance that the cell is occupied. A value of 0.0 indicates that there is a 100 percent chance the cell is unoccupied. A value of 0.5 would indicate an equal probability that the cell is either occupied or unoccupied. Separate variables that provide valuable influence in the force calculation are the number of times that a cell has been sampled by a sensor and the number of times that a percentage of the entire occupancy map has been visited. The usefulness here is to accommodate the dynamic nature of agents moving on the stage as well as the potential for moving targets in the field of regard. As a result, the charge associated with each cell is given by:

$$q_i = \begin{cases} P_O q_{max} & \text{for } 0<V<n \\ 0 & \text{for } V>n \end{cases} \qquad (8)$$

where,

P_O = cell occupancy probability,

q_{max} = maximum possible cell charge,

V = number of visits to the ith cell, and

n = number of times that a percentage of the cells have been visited.

SIMULATION ENVIRONMENT

Cyberbotics' Webots simulation package provides an extremely robust capability and interface with very high resolution characterization of the robots in a physics-enabled, three-dimension, virtual world. The robot agents utilized are the e-puck robot originally developed by the Ecole Polytechnique Federale de Lausanne (EPFL) in Lausanne, Switzerland. E-puck is principally an educational robot that offers significant research capability both through its own autonomous processing and sensing capability, as well as its close linkage with Webots. E-puck is approximately 7 cm in diameter and has a number of sensors including eight infra-red sensors positioned around the robot's perimeter that provide both proximity sensing and light sensing, a 640x480 pixel camera on-board that can provide video and imagery in either color or black and white modes, three omni-directional microphones for sound localization, a three-axis accelerometer, and on-board speaker. Communications is provided by Bluetooth. Programming is in C and compiled via a GNU GCC compiler system.

The representative simulation and experimental agent arena for this research consists of a 120 cm by 240 cm stage bordered by a low wall. The stage is sub-divided into a representative occupancy mapping grid consisting of 1 cm by 1 cm cells each having an associated vector representing dynamic characteristics such as the probability of occupancy or potential charge. The green and white checkerboard appearing in the simulation's stage is physically 10 cm x 10 cm and exists merely to identify rough approximation of localization. The stage also hosts obstacles of varying shapes and sizes as well as identified agent starting points. In simulation, the localization of each agent is provided by a surrogate global positioning system function afforded through the Webots simulation software and in contrast to the real e-puck's lack of such capability. In keeping with the representation and standard established in the Webots simulation package, the world coordinate system is represented as illustrated in the figure below.

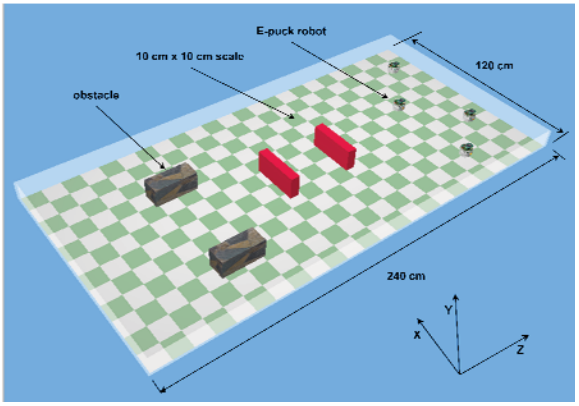


FIGURE 2 – E-puck stage in Webots

In order to visualize and track the progress of the simulation run and its representative evaluation metrics, a MATLAB figure is generated every 100 time steps that offers information on the e-pucks' track history, cell charge, occupancy mapping and cell visit count (Figure 3).

RESULTS

The payload capabilities at this stage in the research are restricted to the proximity sensors. Future work will add the e-puck's camera imaging and video. The simulated stage replicates that forecast for the experiment and consists of a bounded 120 cm by 240 cm flat arena with an unknown number of obstacles of unknown size, shape, and color. The scenario used is to have the MAS search the bounded stage and populate the occupancy map with information collected by the proximity sensing. The scenario is run for a constant 2,000 time steps per simulation experiment. The independent variable considered is the number of agents (e-pucks), from one to four, comprising the multi-agent system. The dependent information collected, reduced, and evaluated are the resultant occupancy maps, cell charge mapping; cell visit count; and map coverage count (Figure 3). The data itself is represented in three-dimensional graphs with dimensional scaling on the x and z axes (the horizontal plane in the graph), and values vertically. The range on the value axis is the same for all four charts to facilitate qualitative analysis, i.e., the charts are all proportionally similar.

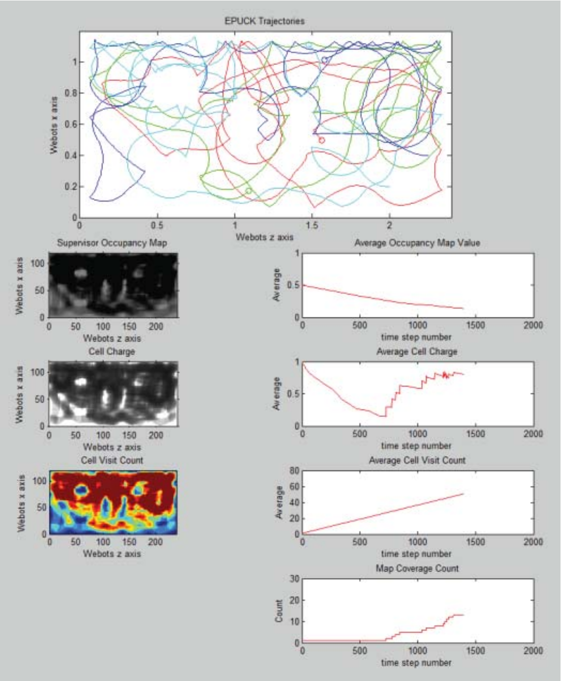


FIGURE 3 – Simulation Metrics Output Screen

There are several very interesting observations and characterizations derived from the reduced and analyzed data.

1. The map coverage count increases non-linearly with additional robots in the multi-agent system. Figure 4 shows the map coverage count witnessed from the simulation runs of the various sized MASs. This is an expected result given the time/space efficiencies offered by MAS in consideration of inter-robot repulsive forces included in the path planning algorithm.

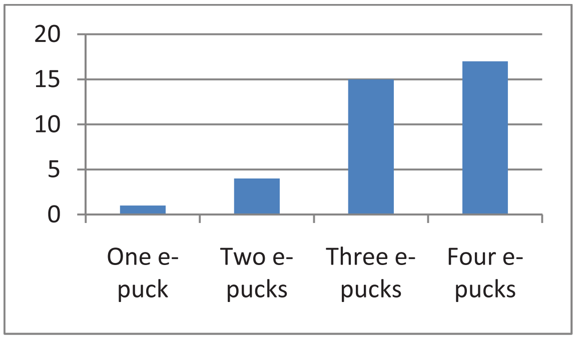


FIGURE 4 – Map Coverage Count

2. The map coverage count curve of a fixed size MAS approaches an asymptote. This is an unexpected result as one would expect the map coverage count to continue increasing as the robots continue to search the space. One observation however is that this is a local asymptote whereby continuing the search would show another exponential growth of the map coverage count to a second asymptote. Alternately, this could very well be an indication that the space has been exhaustively searched such that a confidence could be declared. This phenomenon will be a focus of future research.

3. The potential energy function offers efficient and uniform coverage capability in a search mode. The data represented proves to be uniform over the stage space and comprehensively covers the stage itself. This is quite expected given the influence that occupancy mapping has and the related magnification of unexplored space therein. The one exception noted is discussed in the next item.

4. Obstacles, more specifically the charge mapping of occupancy cells comprising the obstacle space, skew the path planning. This was a shortcoming noted in published research [14] and reconfirmed herein. This is a result of the fact that the robot can never sense inside the perimeter boundary of the obstacles such that the cell charges within are never decremented.

5. The relative proportions of the elements comprising the force calculation in the path planning routine can vary significantly over time. This too is a subject ripe for further investigation in continuing research. As an example, the force associated with a mature occupancy mapping (i.e., a well known environment) is possibly only 1-5 percent of that associated with an unknown environment. In addition, the collective repulsive force from robot to robot is a function of distance between the robots, but is consistent over time since the average distance remains somewhat constant. The result of these two facts is that the influence of robot repulsion over time effectively increases. In this limited case, it actually becomes a desirable feature as one would endeavor to spend less energy searching known space. However, future research will closely consider this phenomenon especially as it applies to the addition of imaging and video payloads and the dynamic nature of their information contribution.

6. The force values drop precipitously as a function of distance. The physics of potential energy functions dictate that the force is inversely proportionate to the square of the distance. In this research, the energy physics were replicated as such. However, Figure 5 shows that this tends to greatly magnify the influence of proximate cells and greatly diminish distance cells. The conclusion drawn here is that in an effort to normalize cell charges, and gain better responsiveness over the entirety of the stage versus proximate cells, research should explore varying the distance's exponent. Clearly, this deviates from classic potential theory and defies the laws of physics, but might nonetheless offer some interesting advantages in path planning efficiencies.

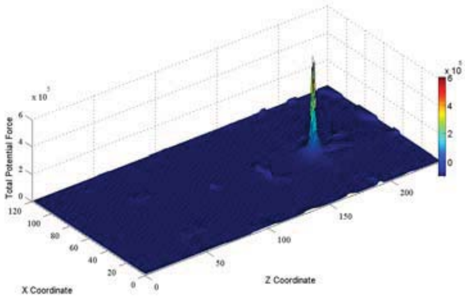


FIGURE 5 – Total Potential Force

7. Transition from global path planning control to local collision avoidance control exhibits undesirable oscillations. Global path planning is generated using the potential function while local collision avoidance, triggered by obstacle detection, uses classic Braitenberg theory. Simulation runs illustrate that as an individual robot transitions between these two control states, an undesirable oscillation appears at the boundary (refer to the upper edge of the stage in Figure 6). This phenomenon results from two realities. First, the cell charges tend to be higher at the perimeter boundaries due in fact to this oscillation. Therefore, the robots strive toward that boundary. Secondly, the existing Braitenberg algorithm is tuned to simply motion reflect off of the boundary as opposed to perimeter follow. This too is an area that will be refined in subsequent research by exploring modification of the Braitenberg matrix, tuning factors that can be incorporated into the transition space, and cell charge calculations at obstacle and perimeter boundaries.

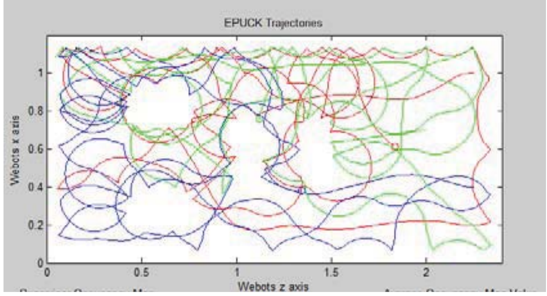


FIGURE 6 – Individual E-puck Trajectories

CONCLUSIONS

This paper explores the characterization of a multi-agent system through novel application of a potential force function that includes tuning coefficients. Initial excursions were run in simulation that varied the number of robot agents teamed to accomplish an area search mission with no a priori target or obstacle information. Data reduction and evaluation shows interesting behaviors for global path planning and local collision avoidance and set the stage for further investigation of potential function coefficient tuning.

ACKNOWLEDGEMENT

The research reported in this paper was supported by the U.S. Army Research Office under grant number W911NF-08-0106.

REFERENCES

[1] Garg, Devendra, and Young, Reed: Coordinated Control of Cooperating Robotic Manipulators, *International Journal of Systems Science*, Vol 21, Issue 11, November 1990 , pp. 2161 - 2176.

[2] Young, Reed: The Coordination of Multiple Robotic Manipulators, Master's thesis, Duke University, 1987.

[3] Parker, Lynn: Multiple Mobile Robot Systems, *Springer Handbook of Robotics*, 2008, pp. 921-941.

[4] Burgard, Wolfram; Moors, Mark; Stachniss, Cyrill; and Schneider, Frank: Coordinated Multi-robot Exploration, *IEEE Transactions on Robotics*, Vol. 21, No. 3, June 3, 2005, pp. 376-386.

[5] Baghaei, Khashayar; and Agah, Arvin: Task Allocation Methodologies for Multi-Robot Systems, *Technical Report ITTC-FY2003-TR-20272-01*, November 2002.

[6] Parker, Lynn: Guest Editorial, Special Issue on Heterogeneous Multi-robot Systems, *Autonomous Robots*, Vol 8, Issue 3, June 1, 2000, pp. 207-208.

[7] Parker, Lynn: The Effect of Heterogeneity in Teams Of 100+ Mobile Robots, *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2003 International Workshop on Multi-robot Systems*, Vol II, March, 2003, pp. 205-218.

[8] Potter, Mitchell; Meeden, Lisa; and Schultz, Alan: Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergence of Specialists, *International Joint Conference On Artificial Intelligence*, Vol 17, Part 1, 2001, pp. 1337-1343.

[9] Grabowski, Robert; Navarro-Serment, Luis; Paredis, Christiaan; and Khosla, Pradeep: Heterogeneous Teams of Modular Robots for Mapping and Exploration, *Autonomous Robotics*, Vol 8, Number 3, June, 2000, pp. 293-308.

[10] Desai, Jaydev; Ostrowski, Jim; and Kumar, Vijay**:** Controlling Formations of Multiple Mobile Robots, *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, Vol 4, May 1998, pp. 2864-2869.

[11] Alur, Rajeev; Das, Aveek; Esposito, Joel; Fierro, Rafael; Grudic, Gregory; Hur, Yerang; Kumar, Vijay; Ostrowski, James; Pappas, George; Southall, B., Spletzer, John; and Taylor, Camillo: A Framework and Architecture for Multirobot Coordination, *Experimental Robotics VII*, 2001, pp. 303-322.

[12] Kim, Jin; Shim, David; Rashid, Shahid; and Sastry, Shankar: Hierarchical System for Multiple-agent Scenarios, *DTIC Report A794314*, January 1, 2002.

[13] Dias, M. Bernardine; Browning, Brett; Veloso, Manuela; and Stentz, Anthony: Dynamic Heterogeneous Robot Teams Engaged in Adversarial Tasks, *Technical Report CMU-RI-TR-05-14*, Robotics Institute, Carnegie Mellon University, April, 2005.

[14] Scott, Adele; and Yu, Changbin: Cooperative Multi-agent Mapping and Exploration in Webots, *Proceedings of the 4th International Conference on Autonomous Robots and Agents, Canberra Australia,* Feb 2009, pp.56-61.

A.2.12 Aggregation of Heterogeneous Units in a Swarm of Robotic Agents

The following paper (on the next 6 pages) was co-authored by Manish Kumar and Devendra P. Garg, and appeared in the Proceedings of the 4th International Symposium on Resilient Control Systems on pages 107–112, published and presented in August 2011.

Aggregation of Heterogeneous Units in a Swarm of Robotic Agents

Manish Kumar
School of Dynamic Systems
University of Cincinnati
Cincinnati, OH 45242
manish.kumar@uc.edu

Devendra P. Garg
Department of Mech. Engg. and Matls. Sci.
Duke University
Durham, NC 27708
dpgarg@duke.edu

Abstract

Formation of patterns in a system of interacting units of heterogeneous types is a self-organized behavior which is seen in many biological systems. Earlier research in this area has indicated that such pattern formation behaviors in biological cells and tissues are made possible because of difference in the adhesivity between different types of cells or tissues. Inspired by this differential adhesivity model, in our earlier research, we had presented a decentralized approach based on differential artificial potential to achieve the segregation behavior in a swarm of heterogeneous robotic agents in which agents of different types formed spatially separate clusters. In this paper, we extend that work by presenting an approach to achieve aggregation in which agents of different types get uniformly mixed with each other. The method is based on the proposition that agents of different types experience varying magnitude of potential while interacting with the agents of different types. An analysis of the system with the proposed approach in Lyapunov sense is carried out for stability. Extensive simulation studies and numerical analysis suggest that the proposed method would lead a population of heterogeneous agents to an aggregated configuration.

1. Introduction

Control of multiple autonomous vehicles collaboratively carrying out a task has received attention of several researchers working in the area of mobile robotics because of its potential to achieve unprecedented performance in terms of efficiency, ability to carry out complex tasks, and robustness. Research in this area has been motivated by applications in a number of fields including cooperative search and rescue operation, surveillance, reconnaissance, and boundary protection. One of the very first applications of formation control of multiple agents was behavioral simulation of flocks of birds, herds of animals and schools of fish for computer graphics by Reynolds [17]. Drawing inspiration from Reynolds' approach, many researchers have focused on designing decentralized controller for achieving flocking behavior. The examples include behavior-based methods [1], leader-follower technique [3, 4], method based on formation constraint and virtual leaders/beacons [5, 13], and Lyapunov function based methods [13, 14, 15, 16]. The concept of artificial potential has been used in robotics by many researchers. For example, artificial potential has been used for path planning [24], manipulator control [23], robot navigation [9], obstacle avoidance [8], and multi-robot formation control [10, 12, 13, 15].

Obtaining a desired shape and pattern of the formation can be critical for a mission relying on coordinated action by multiple mobile agents. In many situations, it may not be possible to integrate all the capabilities, sensing or actuation, required for different kinds of tasks in an individual robot. Accordingly, the robots may have heterogeneous abilities for sensing and actuation that will enable them to perform specific tasks. Heterogeneous robots must be able to self-organize themselves in a mission specific manner to carry out tasks assigned to them. The main contribution of this paper is in the synthesis and analysis of a controller that can be applied to a swarm consisting of two types of robots so that the swarm forms a pattern of homogeneous mixture of robots in which nearest neighbors of a robot of a certain type are robots of the other type. In our earlier papers [11, 12], we presented control algorithms for achieving segregation in which the two populations form spatially separate clusters. In this paper, we extend that work by presenting an algorithm that will accomplish the aggregation in which the two poulations get

978-1-4244-9293-0/11/$26.00 ©2011 IEEE

intermixed homogeneously.

2. Segregation, Aggregation, and Pattern Formation in Biology

Complex patterns form in living systems as a result of interactions among basic constituent units and random processes. Examples include formation of ocular dominance stripes in visual cortex of cat and monkey, and formation of tentacle pattern in hydra. Of particular relevance to current work is the pheomenon of sorting which is seen in several biological systems. Examples include sorting of cells based on their types and functionalities. Cell sorting is one of the basic phenomenon which leads to formation of patterns and organs in living organisms. Most of the strategies or models in literature that can explain formation of patterns rely on differential attraction/inhibition. For example, Swindale's model [19] accounts for formation of ocular dominance stripes in visual cortex based on local activation and lateral inhibition mechanism [6] for like type of synapses, and local inhibation and lateral activation [2, 19], the reverse, for the unlike type of synapses. One of the most popular models is the Turing model [22] which is reaction-diffusion model to mathematically represent the transport phenomenon in biological and natural systems. This model tries to explain the interaction of particles with the environment and their motion in space. In early 1990s, Graner and Glazier [7] proposed a lattice based modified version of large-Q Potts model with differential adhesivity to explain and simulate the sorting of a mixture of two types of biological cells. In fact, it has been long known [18] that it is the difference in intercellular adhesivity that leads to sorting in certain kinds of cellular inetractions while homogeneous intermixing in other kinds. The final state of cell configuration is achieved when the overall surface energy is globally minimized. Based on this principle, Steinberg [18] postulated that two types of cellular units A and B are aggregated when:

$$W_{AB} \geq \frac{(W_{AA} + W_{BB})}{2} \tag{1}$$

where W_{AA} and W_{BB} represent the work of cohesion between particles or cells of same types (i.e., between types A & A, and B & B respectively), and W_{AB} represents the work of adhesion between cells of types A and B. When the above condition is satisfied, the overall surface energy is minimzed when A and B units are alternately arranged, or in other words, the two populations are intermixed. The method for aggregation in artificial mobile agents presented in this paper is motivated by this differential adhesivity phenomenon observed in biological systems.

3. Problem Formulation

The group of mobile agents consists of N fully actuated agents, each of whose dynamics is given by the double integrator:

$$\begin{aligned} \dot{q}_i &= p_i \\ \dot{p}_i &= u_i(t) \qquad i = 1, 2, \ldots, N \end{aligned} \tag{2}$$

where q_i and p_i are m-dimensional position and velocity vectors respectively of agent i. The group of mobile agents, considered in this paper, consists of two different types of agents: type A and type B. The number of agents of type A is N_A and that of type B is N_B such that $N = N_A + N_B$. The objective of this paper is to synthesize a controller that drives the whole swarm to asymptotically flock in a manner that the swarm forms a homogeneous mixture of robots of type A and type B, i.e., nearest neighbors of a robot of type A are robots of type B (and vice-versa). This emergent behavior of formation of homogeneous mixture is referred to in this paper as *aggregation*. Agents are said to flock (asymptotically) when all agents achieve the same velocity vector, distances between the agents are stabilized, and no collisions occur. In order to define the aggregation mathematically, we can define metrics for each agent i which represent average distances between that agent and its neighbors (set of agents N_i within a range of R_{NN}) of each type. Hence, for each agent i of type $X \in \{A, B\}$, we define metrics:

$$R_i^{XY} = \frac{1}{|N_i^Y|} \sum_{j \in N_i^Y} \|q_i - q_j\| \tag{3}$$

where $Y \in \{A, B\}$, and N_i^Y is the set of agents of type Y in N_i. Furthermore, we define:

$$r_{avg}^{AA} = \frac{1}{N_A} \sum_{i \in A} R_i^{AA} \tag{4}$$

$$r_{avg}^{BB} = \frac{1}{N_B} \sum_{i \in B} R_i^{BB} \tag{5}$$

$$r_{avg}^{AB} = \frac{1}{N_A + N_B} \sum_{i \in A \cup B} R_i^{AB} \tag{6}$$

Now *Aggregation* is defined as a configuration of agents where the average distance between the agents of like types (type A or type B) is more than the average distance of agents between the unlike types (between agents of type A and type B). Alternatively,

$$r_{avg}^{AA} > r_{avg}^{AB}, \quad r_{avg}^{BB} > r_{avg}^{AB} \tag{7}$$

The above condition is based upon the average proximity of agents of different types in the defined neighborhood of a given agent. Another metric can be based upon the distance to the nearest neighbor of each type from a given agent. The distance of the nearest neighbor of agent i of type $X \in \{A,B\}$ is given by:

$$R^{XY}_{iNN} = min_{j \in N_i^Y} \left\| q_i - q_j \right\| \tag{8}$$

Based upon the above equation, the metric can be defined as:

$$r^{AA}_{avg(NN)} = \frac{1}{N_A} \sum_{i \in A} R^{AA}_{iNN} \tag{9}$$

$$r^{BB}_{avg(NN)} = \frac{1}{N_B} \sum_{i \in B} R^{BB}_{iNN} \tag{10}$$

$$r^{AB}_{avg(NN)} = \frac{1}{N_A + N_B} \sum_{i \in A \cup B} R^{AB}_{iNN} \tag{11}$$

$$r^{BA}_{avg(NN)} = \frac{1}{N_A + N_B} \sum_{i \in A \cup B} R^{BA}_{iNN} \tag{12}$$

And the condition for aggregation can be given by:

$$r^{AA}_{avg(NN)} > r^{AB}_{avg(NN)}, \quad r^{BB}_{avg(NN)} > r^{BA}_{avg(NN)} \tag{13}$$

4. Control Law Formulation

This section presents the control law which causes a population of heterogeneous agents to asymptotically flock as well as aggregate. For a system of N mobile agents with N_A agents of type A and N_B agents of type B, the following feedback control law is considered:

$$u_i = -\sum_{j \in N_i} \nabla_{q_i} V_{ij}(\|q_j - q_i\|) - a \sum_{j \in N_i} (p_i - p_j) \tag{14}$$

where u_i is the control input to the agent i, N_i is the set of agents in the neighborhood of agent i, $V_{ij}(\|q_j - q_i\|)$ is the artificial potential of interaction between agents i and j, $\|q_j - q_i\|$ is the norm of vector $(q_j - q_i)$ representing the Euclidean distance between agent i and j, and ∇_{q_i} is the gradient with respect to coordinates of agent i i.e., q_i. First term in Equation (14) represents the gradient of potential function, and the second term represents damping and causes the agents to match their velocities with each other. The artificial potential is a non-negative function of relative distances between a pair of neighbors given by $V_{ij}(q_i, q_j) : R^{2m} \to R_{\geq 0}$. Artificial potential function, V_{ij} due to interaction between two agents i and j can be expressed [13] as:

$$V_{ij} = a \left(ln(q_{ij}) + \frac{d_0}{q_{ij}} \right) \tag{15}$$

where, a is a scalar control gain, and $q_{ij} = \|q_j - q_i\|$. The parameter d_0 represents the inter-agent distance below which (i.e. when $q_{ij} < d_0$) the interaction force is repulsive (negative) and above which (i.e. when $q_{ij} > d_0$) the interaction force is attractive. Figure 1 shows the potential function plotted against the inter-agent distance. As indicated in the figure, the potential becomes minimum when the inter-agent distance is d_0. The initial condition and dynamics exclude the situations where $q_i = q_j$ where the control law (14) is undefined.

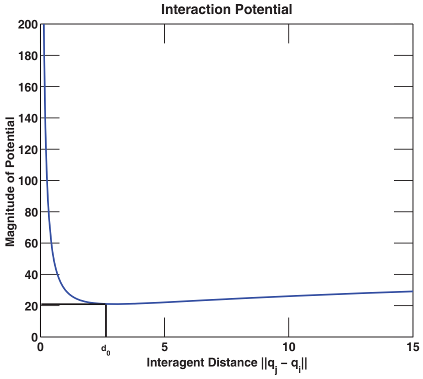


Figure 1. Interaction Potential versus Inter-Agent Distance

Since there are two types of mobile agents involved in the system, there are three different kinds of artificial potentials involved: a) Potentials arising due to interactions between types A and A (V^{AA}_{ij}), b) Potentials arising due to interactions between types B and B (V^{BB}_{ij}), and c) Potentials arising due to interactions between types A and B (V^{AB}_{ij}). Our main result for cause of aggregation is based on the concept of differential potential, i.e., agents experience different magnitudes of potential when they are interacting with the agents of different types. This is achieved by assuming different values of the parameter d_0 for different types of interaction potentials. Explicitly, in Equation (15),

$$d_0 = \begin{cases} d_0^{AA} = d_0^{BB} & \text{if} (i,j \in A) \text{or} (i,j \in B) \\ d_0^{AB} & \text{if} (i \in A, j \in B) \text{or} (i \in B, j \in A) \end{cases} \tag{16}$$

The control law for aggregation can be achieved when:

$$d_0^{AA} = d_0^{BB} > d_0^{AB} \tag{17}$$

Figure 2 shows the plot of force of interaction due to similar types and due to dissimilar types of robots versus inter-agent distance when the condition (17) for segregation controller is met. In this case, it can be seen that the interaction force between agents of same types is lesser than that between agents of different types at any given distance. Hence, this method of aggregation, based on differential potential, is analogous to Steinberg's [18] explanation of cellular aggregation based on differential adhesiveness (see Equation (1)).

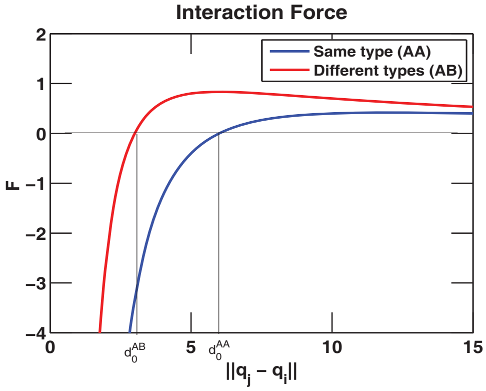


Figure 2. Interaction Force between Agents

5. Stability Analysis

In this section, we carry out an analysis of convergence and stability properties of the system of multiple agents obeying dynamics given by Equation (2) under control law given by Equation (14). In order to carry out the stability analysis of the collective motion of the agents, the following positive definite function can be chosen as the Lyapunov function:

$$\phi(\mathbf{q},\mathbf{p}) = V(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{p} \tag{18}$$

where $\mathbf{q} \in R^{mN}$ is stacked position vector of all agents, $\mathbf{p} \in R^{mN}$ is stacked velocity vector of all agents, and $V(\mathbf{q}) : R^{mN} \to R_{>0}$ is the total potential energy of the system which consists of three parts resulting from interactions between agents of: 1) type A, 2) types A and B, and 3) type B. This can be written as:

$$\begin{aligned} V(\mathbf{q}) &= \mathbf{V_{AA}}(\mathbf{q}) + \mathbf{V_{BB}}(\mathbf{q}) + \mathbf{V_{AB}}(\mathbf{q}) \\ &= \frac{1}{2}\sum_{i\in A}\sum_{j\in A, j\neq i} V_{ij}(\|q_j - q_i\|) + \sum_{i\in A}\sum_{j\in B} V_{ij}(\|q_j - q_i\|) \\ &+ \frac{1}{2}\sum_{i\in B}\sum_{j\in B, j\neq i} V_{ij}(\|q_j - q_i\|) \end{aligned} \tag{19}$$

The collective dynamics of the system is given by:

$$\dot{\mathbf{q}} = \mathbf{p} \tag{20}$$

$$\dot{\mathbf{p}} = -\nabla V(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{p} \tag{21}$$

where $\hat{L}(\mathbf{q}) \in R^{mN\times mN}$ is m-dimensional graph Laplacian (see reference [15]). Graph Laplacian represents the interconnections present in a graph in a matrix form, and can be obtained from other graph theoretic quantities such as degree matrix and adjacency matrix. Among other important properties of graph Laplacian matrix $\hat{L}(\mathbf{q})$, it is a positive semi-definite matrix.

Lemma 5.1. *Consider a system of N mobile agents. Each of the agents follows the dynamics given by Equation (2), and with feedback control law given by Equation (14). For any initial condition belonging to the level set of $\phi(\mathbf{q},\mathbf{p})$ given by $\Omega_C = \{(\mathbf{q},\mathbf{p}) : \phi(\mathbf{q},\mathbf{p}) \leq C)\}$ with $C > 0$, and when the underlying graph of the system is fully connected all the time, then the system asymptotically converges to the largest invariant set in $\Omega_I \subset \Omega_C$. The points in the largest invariant set in Ω_I have a velocity that is bounded, the velocities of all agents match, and the total potential of all agents given by Equation (19) approaches a local minimum.*

Proof. Differentiating $\phi(\mathbf{q},\mathbf{p})$ with respect to time and using Equation (21) one gets:

$$\begin{aligned} \dot{\phi}(\mathbf{q},\mathbf{p}) &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\dot{\mathbf{p}} \\ &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T(-\nabla V(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{p}) \\ &= -\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p} \\ &= -\sum_i\sum_j \|p_j - p_i\|^2 \leq 0 \end{aligned} \tag{22}$$

From Lasalle's Invariance Principle, all solutions of the system starting in Ω_C will converge to the largest invariant set in $\Omega_I = \{(\mathbf{q},\mathbf{p}) \in \Omega_C : \dot{\phi}(\mathbf{q},\mathbf{p}) = 0)\}$, and this happens when the velocities of all agents match. For a detailed proof of this, please see references [13, 15]. Furthermore, in the steady state, the velocities of the agents do not change, i.e., $\dot{\mathbf{p}} = 0$. For a proof of this result, see references [20, 21]. From Equation (21), it follows that at the steady state:

$$\nabla V(\mathbf{q}) = 0 \tag{23}$$

Hence, the total potential of all agents is locally minimized. □

6. Simulation Results and Discussions

Extensive simulations were carried out to verify that the proposed controller would lead to aggregation. In the simulations, the following parameters were assumed: $d_0^{AA} = d_0^{BB} = 6$, and $d_0^{AB} = 3$.

Figure 3 shows the configuration of a population of 20 agents (10 each of type A and B) in a 2D space at different times during the simulation. The agents started off at a random configuration, and control law given by Equation (14) based on differential potential was applied to the agents. The final configuration at time T=750 sec shows that the agents of types A and B form a homogeneous mixture.

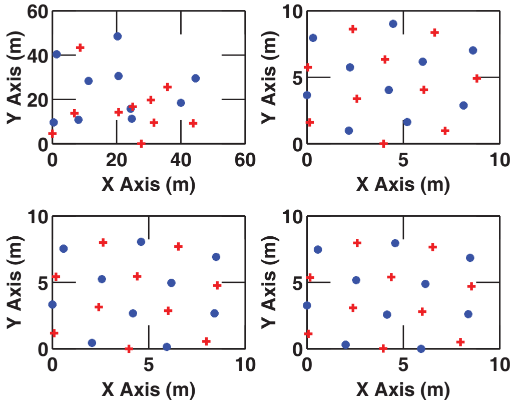


Figure 3. Configurations of Agents at Times T=0 (top left), T=250 sec (top right), T=500 (bottom left), and T=750 sec (bottom right)

Figure 4 shows the plot of average distances between agents of types A and A (r^{AA}_{avg}), B and B (r^{BB}_{avg}), and A and B (r^{AB}_{avg}) versus time for the above simulation. At the final configuration, the average distances r^{AA}_{avg}, r^{BB}_{avg}, and r^{AB}_{avg} were found out to be 4.87, 4.87, and 4.43 respectively, which clearly shows that the population was aggregated based on the condition given by (7). Also, on average: distance of the nearest type A neighbor of type A robot ($r^{AA}_{avg(NN)}$) was 3.12; distance of the nearest type B neighbor of type A robot ($r^{AB}_{avg(NN)}$) was 1.98; distance of the nearest type B neighbor of type B robot ($r^{BB}_{avg(NN)}$) was 3.12; and distance of the nearest type A neighbor of type B robot ($r^{BA}_{avg(NN)}$) was 1.98. Hence, aggregation happens according to condition (13) as well.

The results given above were for just one simulation run. In order to verify that the method presented in this paper leads to aggregation in general in a population of heterogeneous agents, an extensive simulation study was carried out in which 100 runs were performed. Figure 5 shows the average distance between the agents at steady-state. In each of the simulation runs, the population of agents consisting of types A and B was initialized in a random configuration obtained via uniform distribution of agents in 2D space, and number of agents of type A and B were each chosen randomly between values 5 and 15. Each of the runs was carried out for

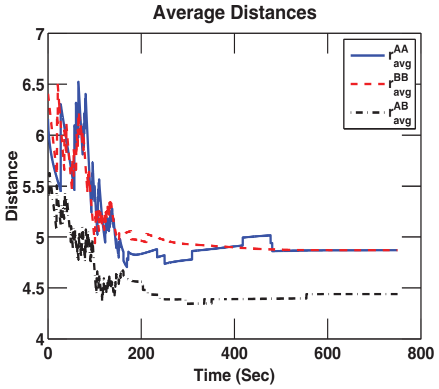


Figure 4. Average Distances Between Agents of Types A and A (r^{AA}_{avg}), B and B (r^{BB}_{avg}), and A and B (r^{AB}_{avg})

750 seconds of simulation time. The average distances between agents shown in the figure are calculated at the steady (final) state. It can be easily seen that the average distance between agents of type A (r^{AA}_{avg}) and average distance between agents of type B (r^{BB}_{avg}) is larger than the average distance between agents of type A and B (r^{AB}_{avg}) for each of the simulation runs. Also, the plot of average distances to respective nearest neighbors for each types of agents for all simulation runs are shown in Figure 6. This figure also demonstrates that aggregation happens for each simulation run according to condition (13).

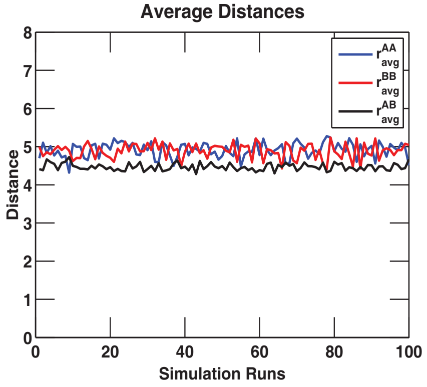


Figure 5. Average Distance between Agents

7. Conclusions

The paper presents a decentralized method to achieve self organized bahavior of aggregation in a population of heterogeneous agents. In this behavior, the heterogeneous agents form a homogeneous mixture.

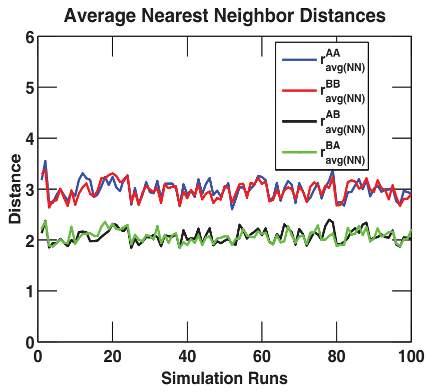


Figure 6. Average Nearest Neighbor Distance between Agents

The method is based on the concept of differential artificial potential. The paper presents the stability analysis of a population of agents in Lyapunov framework, and lays down an analytical foundation for synthesis of controllers for aggregation in artificial potential function framework. Extensive simulation studies verify the results obtained in this paper, and show the effectiveness of the proposed method in achieving the aggregation behavior in a swarm of heterogeneous agents.

References

[1] T. Balch and R. Arkin. Behavior-based formation control for multirobot systems. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[2] E. Bonabeau. From classical models of morphogenesis to agent-based models of pattern formation. *Artificial Life*, 3:191–211, 1997.

[3] J. P. Desai, J. P. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2864–2869, 1998.

[4] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[5] M. Egerstedt and X. Hu. Formation constrained multi-agent control. In *IEEE International Conference on Robotics and Automation*, pages 3961–3967, 2001.

[6] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12:30–39, 1972.

[7] F. Graner and J. A. Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical Review Letters*, 69(13):2013–2016, Sept. 1992.

[8] J. O. Kim and P. K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349, 1992.

[9] D. E. Koditschek. Exact robot navigation by means of potential functions: some topological considerations. In *IEEE Conference on Robotics and Automation*, volume 4, pages 1–6, 1987.

[10] M. Kumar, K. Cohen, and B. HomChaudhuri. Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires. *Journal of Aerospace, Computing, Information, and Computation*, 8(1):1–16, 2011.

[11] M. Kumar, D. P. Garg, and V. Kumar. Self-sorting in a swarm of heterogeneous agents. In *American Control Conference*, pages 117–122, 2008.

[12] M. Kumar, D. P. Garg, and V. Kumar. Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Transactions on Automatic Control*, 55(3):743–748, 2010.

[13] N. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *IEEE International Conference on Decision and Control*, pages 2968–2973, 2001.

[14] P. Ogren, M. Egerstedt, and X. Hu. A control Lyapunov function approach to multi-agent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.

[15] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, Mar. 2006.

[16] R. Olfati-Saber and R. M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *IEEE Conference on Decision and Control*, volume 3, pages 2965–2971, 2002.

[17] C. Reynolds. Flocks, birds, and schools:a distributed behavioral model. *Computer Graphics*, 21:25–34, 1987.

[18] M. S. Steinberg. Reconstruction of tissues by dissociated cells. *Science*, 141:401–411, 1963.

[19] N. V. Swindale. A model for the formation of ocular dominance stripes. *Philosophical Transactions of the Royal Society of London B*, 208:243–264, 1980.

[20] H. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *IEEE International Conference on Decision and Control*, pages 2010–2015, 2003.

[21] H. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, 2007.

[22] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237(641):37–72, 1952.

[23] R. Volpe and P. Khosla. Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1423–1436, 1990.

[24] C. W. Warren. Global path planning using artificial potential fields. In *IEEE Conference on Robotics and Automation*, pages 316–321, 1989.

A.2.13 On the Stability of Swarm Consensus under Noise Control

The following paper (on the next 8 pages) was co-authored by Gregory K. Fricke, Bruce Rogers, and Devendra P. Garg, and appeared in the Proceedings of the 4th ASME Dynamic Systems and Control Conference, Volume 1, on pages 291–298, published and presented in October 2011.

Proceedings of the ASME 2011 Dynamic Systems and Control Conference
DSCC2011
October 31 - November 2, 2011, Arlington, VA, USA

DSCC2011-6061

ON THE STABILITY OF SWARM CONSENSUS UNDER NOISY CONTROL

Bruce Rogers
Dept. of Mathematics
Duke University
Durham, North Carolina 27708
bruce@math.duke.edu

Gregory K. Fricke, Devendra P. Garg*
Dept. of Mechanical Engineering & Materials Science
Duke University
Durham, North Carolina 27708
{gregory.fricke, dpgarg}@duke.edu

ABSTRACT

Representation of a swarm of independent robotic agents under graph-theoretic constructs allows for more formal analysis of convergence properties. We consider the local and global convergence behavior of an N-member swarm of agents in a modified consensus problem wherein the connectivity of agents is governed by probabilistic functions. The addition of a random walk control ensures ε-stability of the swarm consensus. Simulation results are given to show the rate of convergence to consensus and planned experiments are described.

1 Introduction

Control of a swarm of robots may be achieved in many different ways. The seminal work of Reynolds [1] gave the first algorithmically efficient representation of a flock with very simple rules. Reynolds' work was ground-breaking, allowing simulation of the group dynamics involving very large numbers of members. Interestingly, the fundamental work of Braitenberg [2] in the area of behavioral robotics predates Reynolds' research, though that work was focused mostly on individual robots or robot pairs. This idea was further developed by additional research in swarm robotics by Matarić [3] and Parker [4]. Early work was quite ad-hoc, but behavioral control continues to find an increasingly rigorous mathematical framework. For variations on this topic, see, e.g., [5–14].

Swarming laws may be applied to a variety of cooperative-robotics settings. Each of these settings, though, has a primary goal of achieving *agreement* or *consensus*. The specific meaning of agreement in each scenario is different, but in general the implication is that the states of each robot evolve in a coordinated way to achieve some goal. A few specific types of agreement are creation and maintenance of formation [7, 13–15], robot rendezvous [16], plume localization [17], and robot segregation [18]. In this paper, we are concerned with rendezvous and formation control.

Formation control may be accomplished under several control regimes. Clearly, a centralized controller could drive robots to a desired configuration under straightforward trajectory control. It is much more difficult and interesting to develop such capability under a distributed control paradigm, allowing greater (if not unlimited) scalability of the swarm. Distributed control of formations falls into the control regimes of potential functions, geometric control, and graph-based control. Several additional examples of distributed formation can be found in the works of Yun [19], Balch [7], Yamaguchi [20], and Antonelli [21].

In Section 2, we review the basic notation of graph-based control and offer a control framework based on potential field functions, which are used to balance the relative importance of control goals such as obstacle avoidance, goal-seeking, and robot following (or avoidance) [12, 22–27]. Section 3 provides a proof of asymptotic convergence using the potential field framework, and Section 4 uses simulations to explore the rates of convergence. We present some planned experiments in Section 5 and our conclusions in Section 6.

2 Graphical Representation

A brief discussion of graphs is in order. Graph notation varies in the literature; the following notation will be used in this paper. An undirected graph $\mathcal{G} = \mathcal{G}(V, E)$ of order N consists of a set of N nodes or vertices $V_{\mathcal{G}} = \{1, 2, ..., N\}$ and a set of connections or edges $E_{\mathcal{G}} \subset \{V_{\mathcal{G}} \times V_{\mathcal{G}}\}$. If a pair of nodes is in the set of edges, $(i, j) \in E_{\mathcal{G}}$, the nodes are *adjacent* and are called *neighbors*, indicated by $i \sim j$ or ij. The set of neighbors of a

*Address all correspondence to this author.

node i is given by $\mathcal{N}_i = \{j \in V_{\mathcal{G}} \mid j \sim i\}$. Note that in this case, loop edges, e.g., $(i,i) \in E_{\mathcal{G}}$, are allowed. When it is clear from the context, the subscript will be dropped, e.g., $V = V_{\mathcal{G}}, E = E_{\mathcal{G}}$, and $\mathcal{G}(V,E) = \mathcal{G}(V_{\mathcal{G}}, E_{\mathcal{G}})$.

The *degree* of a node is the cardinality of $\mathcal{N}_i$, denoted by $\Delta_i = |\mathcal{N}_i|$. The vector of degrees of the nodes of $\mathcal{G}$ is denoted $\Delta\mathcal{G} = [\Delta_1, ..., \Delta_i, ..., \Delta_N]^T$. If every node is connected to every other node except itself (so that $\Delta_i = N-1$ for all vertices i) then graph $\mathcal{G}$ is called complete. A complete graph of order N is also denoted by K_N.

The adjacency matrix representing the neighborhood relations among agents is defined in Eq. (1).

$$[\mathcal{A}]_{ij} = \begin{cases} 1 & i \sim j \\ 0 & otherwise \end{cases}. \tag{1}$$

The adjacency matrix leads to an important data structure for studying aggregation and agreement, the graph Laplacian, defined as:

$$\mathcal{L}(\mathcal{G}) = \mathbf{diag}(\Delta\mathcal{G}) - \mathcal{A}. \tag{2}$$

Note that the degree vector, $\Delta\mathcal{G}$, is equivalent to the vector of row-sums of $\mathcal{A}$; in this way, the definition is easily extended to weighted networks. Also notice that $\mathcal{L}(\mathcal{G})$ is non-negative definite. The graph Laplacian is intimately related to diffusions on the network, and the eigenvalues and eigenvectors of $\mathcal{L}(\mathcal{G})$ encode geometric information about the graph $\mathcal{G}$. For example, the number of zero eigenvalues of $\mathcal{L}(\mathcal{G})$ corresponds to the number of connected components of $\mathcal{G}$. Additional information found within the spectra of $\mathcal{L}(\mathcal{G})$, $\mathcal{A}$, $\mathcal{L}(\mathcal{G}_w)$, and $\mathcal{A}_w$, are discussed in the 1988 monograph of Cvetković *et al* [28], as well as in [29–35].

Graph-theoretic control of formations is not novel; the reported work of Tanner, Jadbabaie, and Kumar [13,14] expounded on this topic. More recently, an extensive set of research has been conducted regarding the controllability of such swarms under graph-theoretic constructs. Mesbahi published several papers [36–38] on controllability within a swarm for *agreement*, along with several related works by Hatano, Das, Rahmani, Chen, Kim, and Tan relating to the study and manipulation of the graph Laplacian and its spectra for swarm control [15, 39–42]. Of particular focus in the proposed research is the extension of such graph-theoretic controllability concepts to random graphs along the lines of Hatano and Mesbahi [43, 44]. The extension to the dual of controllability, observability, has additionally been explored by Mesbahi and Zelazo [45, 46].

The key element in the works of Mesbahi is the notion of the state-dependent graph. In general, suppose each of the N agents sits in an n-dimensional state space, so that agent i's state is $x_i \in \mathbb{R}^n, i = 1, \ldots, N$. For example, the state vector x_i may have coordinates determining agent i's position and momentum. We denote the collective state of the system $x = [x_1, x_2, \ldots, x_N]T$, and $X \subset \mathbb{R}^{Nn}$ is the set of possible states. The state dependent graph mapping $X \to g(X)$ sends each configuration of system states to a graph $g(X)$. The agents form the vertices of $g(X)$ and (i,j) is an edge if and only if $\|x_i - x_j\|_2 < \rho$, for some $\rho > 0$ and $i \neq j$. The neighborhood of agent i in $g(X)$, written $\mathcal{N}_i(X)$, is state dependent. The 2-norm has an obvious physical meaning that is directly applicable to real connectivity problems such as wireless networking (where ρ indicates the communication range), or a laser rangefinder (or rangefinder pair) with full 2π angular coverage or an array of vision sensors (where ρ is the detection range for these sensors). Alternatively, any $p-$norm may be used in a similar fashion, though with somewhat lower physical meaning.

The goal is to develop a distributed control which forces the agents to aggregate. In Eq. (3) and Eq. (4) below, a swarming law is given using the framework of state-dependent graphs. In general, let $x_i(t)$ be the n-dimensional state of agent i at time-step t, and define the updating of $x_i(t)$ by $n-$dimensional, first-order dynamics:

$$x_i(t+1) = x_i(t) + u_i(t) \tag{3}$$

$$u_i(t) = A\Psi(t) - Bf(\mathcal{N}_i(X_t), X_t) \tag{4}$$

Here, $u_i(t)$ is the self-determined control of agent i at time-step t, and f is some linear function of the states of the neighbors of i at time t that determines the aggregation interaction. In the next section, f can be thought of as the average of the positions of the neighbors of i. The objective function $\Psi(t)$ is independent of the swarm interaction, and the real constants $A, B \geq 0$ are relative scaling gains.

There are several examples in the literature of necessary and sufficient graph theoretic conditions which force agents to consensus [47–49]. We offer a simple distributed control (not a general theoretical framework) that forces agents to consensus. If the communication network depends on position, $A = 0$, and the aggregation function is a local contraction, the agents will tend to clump together in separate clusters. In the next section we show that a random function $\Psi_i(t)$ can force the agents to eventual consensus regardless of initial conditions.

3 Proof of Consensus Stability

Suppose there are N agents distributed in a compact and convex region $D \subset \mathbb{R}^d$ (generally, $d = 2$ or 3). If the communication network is fixed and connected, the agents can steer to the average position of their neighbors and eventually achieve consensus. However, if the communication network depends on the position of the agent and some fixed sensing radius (much smaller than the environment), agents will not, in general, converge to consensus by averaging their position with their neighbors. In this section we show that adding a small random perturbation after the averaging step can force the agents to converge, under the right conditions.

The key conditions are (i) a reflecting boundary in the environment and (ii) the noise to sensing radius ratio. There are two parts to the argument. First, if all the agents can sense each other and the size of the random jump is small enough, they will always be near each other. Second, since there are reflecting boundary conditions, the agents must all be near each other at some time. This result is not in itself groundbreaking, but it is interesting that we can force the agents to converge by adding noise. Also, the averaging-plus-noise scheme is simple to implement in software and actual robot systems. In the rest of this section, we make these arguments more formal.

For $i = 1, \dots N$, $x_i(t) \in D$ and the initial conditions $x_i(0)$ are arbitrary. Let the neighborhood of agent i and time t be $\mathcal{N}_i(t) = \{j \in V : \|x_i(t) - x_j(t)\| < \rho\}$ for some sensing radius $\rho > 0$. Here $\mathcal{N}_i(t) \neq \emptyset$ since $i \in \mathcal{N}_i(t)$, and typically, $R \ll diam(D)$.

The dynamics of the particles are given by

$$x_i(t+1) = \Psi_i(t) + |\mathcal{N}_i(t)|^{-1} \sum_{j \in \mathcal{N}_i(t)} x_j(t). \tag{5}$$

$\Psi_i(t)$ a random perturbation whose measure is supported on an open set $U_i(t)$ containing the origin. Importantly, we must restrict $U_i(t)$ so that the jump $\Psi_i(t)$ cannot take the agent out of the domain D. The restriction can be considered as reflecting boundary conditions. With a little manipulation these dynamics fit the framework of Eq. (3) and Eq. (4). Without the noise term, the agents cannot in general achieve agreement if the initial graph is disconnected. With certain conditions of the random variables $\Psi_i(t)$, once all the particles are close to one another, they stay close. However, their position will not be fixed, which is sometimes called ε-consensus.

The first proposition shows that if the size of the jump is bounded, then when *all* the agents are near each other, they stay close.

Proposition 1. *Taking the dynamics defined by Eq. (5), suppose the Ψ_i^t are i.i.d. random variables drawn from a distribution that is absolutely continuous with respect to Lebesgue measure and supported on some open set $U_i(t)$ containing the origin. If $diam(U_i(t)) < \rho/2$, then for all i and j*

$$\mathcal{N}_i(t) = \mathcal{N}_j(t) = V \Rightarrow \max_{i.j} \|x_i(t+1) - x_j(t+1)\| < \rho.$$

Proof.

$$\begin{aligned}
&\max_{i,j} \|x(t{+}1) - x_j(t+1)\| \\
&\qquad = \max_{i,j} \Big\| \frac{1}{N} \sum_{k=1}^{N} x_k(t) + \Psi_i^t - \frac{1}{N} \sum_{k=1}^{N} x_k(t) - \Psi_j^t \Big\| \\
&\qquad = \max_{i,j} \|\Psi_i^t - \Psi_j^t\| \\
&\qquad \le \max_{i,j} (\|\Psi_i^t\| + \|\Psi_j^t\|) < \rho
\end{aligned}$$

Drawing a random jump from an arbitrary open set $U_i(t)$ may seem overly abstract, but since we have not yet implemented the algorithm in experiment, we are striving for generality. In the next section, we use uniform random variables on a disc. However, the proof allows for *any* bounded distribution, for example, beta or exponential, levy, or normal (with cut-offs). Also, the assumption that the distribution is absolutely continuous with respect to Lebesgue measure on the sets $U_i(t)$ is essential in the proof of Lemma 1 below. We now wish to show that eventually the particles must be close to one another.

Lemma 1. *Taking the dynamics defined by Eq. (5), suppose the Ψ_i^t are i.i.d. random variables drawn from a distribution that is absolutely continuous with respect to Lebesgue measure and supported on some open set $U_i(t)$ containing the origin, and let $E \subset D$ be open. Also suppose that the diameters of all the sets $U_i(t)$ are bounded away from zero. Then there is a positive constant μ and a* deterministic *time T such that for all $i = 1, \dots, N$ and $t \ge 0$*

$$P(x_i(t+T) \in E) > \mu.$$

Proof. Suppose not. Then there is an open $E_0 \subset D$, some agent $\alpha \in V$, and some strictly increasing sequence of times $\{\tau_i\}_{i \ge 1} \subset \mathbb{N}$ such that $P(x_\alpha(\tau) \in E_0) = 0$ for $i \ge 1$. Take $r > 0$ such that U contains the ball of radius r centered at the origin. For $k = 1, 2, \dots$, define the sets $E_{k+1} = \{x \in D : d(x, E_k) < r\}$, and let

$$c_\alpha(t) = |\mathcal{N}_\alpha(t)|^{-1} \sum_{j \in \mathcal{N}_\alpha(t)} x_j(t)$$

be the center of agent α's communication group at each time t. For $k > 0$, if $c_\alpha(t) \in E_k$, then $P(x_\alpha(t+1) \in E_{k-1}) > 0$ since $(c_\alpha + U) \cap E_{k-1}$ is non-empty and open and the Ψ_α^t are drawn from a measure absolutely continuous with Lebesgue measure. So we must have $c_\alpha(\tau_i) \notin E_1$ for all $i \ge 1$; i.e., $P(c_\alpha(\tau_i) \in E_1) = 0$ for $i \ge 1$. Now, each E_k is open, $E_k \subset E_{k+1}$ and D is compact. Thus, there is a finite positive integer F such that $D \subset E_F$. By finite recursion, $P(c_\alpha(\tau_i) \in E_F) = 0$ for all $i \ge 1$, a contradiction.

The above proof just shows that if an agent wanders randomly in D, it eventually returns to the same spot. Theorem 1 puts the two pieces of the argument together by saying that all the agents must eventually be near the same spot at the same time. Then Proposition 1 gives us that they will continue to be close to each other.

Theorem 1. *If diam$(U) < \rho/2$, there is a random time τ such that*

$$P(\max_{i,j} \|x_i(s) - x_j(s)\| < \rho) = 1$$

for all $s > \tau$.

Proof. Take any ball of radius R $B_R \subset D$. By the lemma, there is a positive constant μ and a deterministic time T such that $P(x_i(T) \in B_R) > \mu$ for all $x_i(0)$. Then set

$$\mathbf{1}_i(k) = \begin{cases} 1 & x_i(kT) \in B_R \\ 0 & \text{else.} \end{cases}$$

The $\mathbf{1}_i(k)$ are Bernoulli random variables indexed by $i = 1, \ldots, N$ and $k = 1, 2, \ldots$. Almost surely, there is $k_0 \geq 1$ such that

$$\prod_{i=1}^{N} \mathbf{1}_i(k_0) = 1.$$

Thus set $\tau = k_0 T$ and apply Proposition 1.

3.1 Discussion

The theorem takes advantage of the random walk induced by the noise. In the product space $X = \Pi_{i=1}^{N} D$, there is an invariant set around the diagonal, and the noisy dynamics will eventually force the agents inside the attracting set. Significantly, noisy perturbations considered are quite general, so it's possible to apply the theorem if the noise isn't symmetric about the origin. That is, we can consider shaped noise that is realistic for the sensing capabilities of robots.

The convexity of the space D is never used in the proof. It is needed so that the averages in the agreement algorithm are well-defined. If we're careful about defining line of sight and which agents participate in the averaging, the assumption can be relaxed to a space that is compact with open interior. This allows for spaces with obstacles. If the space is not compact, the agents may wander off in different directions. This can be addressed by giving the agents a notion of "center" of the space so the random walks are recurrent.

4 Simulation

The previous section guarantees convergence to consensus, but the theorem makes no mention about the size of the domain D or the number of agents. Thus, we have no information about the *rate* of convergence. In this section, we simulate agents using the dynamics in Eq. (5) to explore the rate at which the noise drives the agents to consensus. We take the support of the noise process $U_i(t)$ to be a disc of radius R. Without loss of generality, we set the sensing radius $\rho = 1$, and the domain D is a square region in $\mathbb{R}^2$

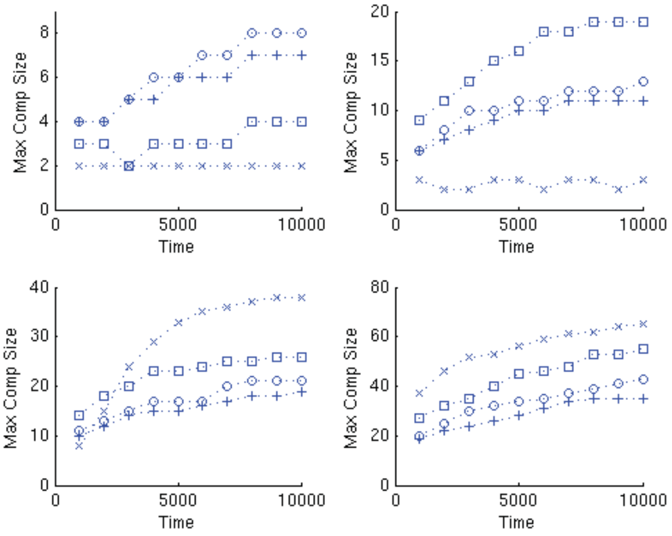


Figure 1. The number of agents in the maximal connected component for simulations in the smallest environment ($W = 20$). The medians of 100 iterations are plotted. Each panel has a fixed number of agents and varies the jump size. Crosses are $R = 0.5\rho$; circles are $R = 0.6\rho$; squares are $R = 0.8\rho$; and x's mark $R = \rho$. Upper left: $N = 10$. Upper right: $N = 20$. Lower left: $N = 40$. Lower right: $N = 100$.

4.1 Uniform initial conditions

The positions of the agents take values uniformly at random in a square region, and three widths of the square, $W = 20, 60$, and 100 are considered. In each of the regions, we also test four sizes of agent population: $N = 10, 20, 40$ and 100. Since the dynamics are random, each parameter setting is run 100 times from initial positions chosen uniformly at random from D, and the dynamics are run in each iteration for 10,000 time steps.

The jumps considered here are determined first by generating uniform random variables $r \in [0, R]$ and $\theta \in [0, 2\pi]$. We set the x-coordinate of the jump to be $r\cos(\theta)$ and the y-coordinate to be $r\sin(\theta)$. Note that this is not a uniform sampling of the disc of radius R; it is biased toward the center.

Also, in Section 3, a triangle inequality is used to show that if the diameter $R < \rho/2$, the noise cannot drive agents out of their mutual sensing radius once they are all close to each other. However, the triangle inequality is a notoriously dull bound, so we also simulate four different sizes of random jumps $R = 0.5\rho, 0.6\rho, 0.8\rho, \rho$ for each of the values of N and W. That is, the smallest jump considered is the one proven to give asymptotic convergence.

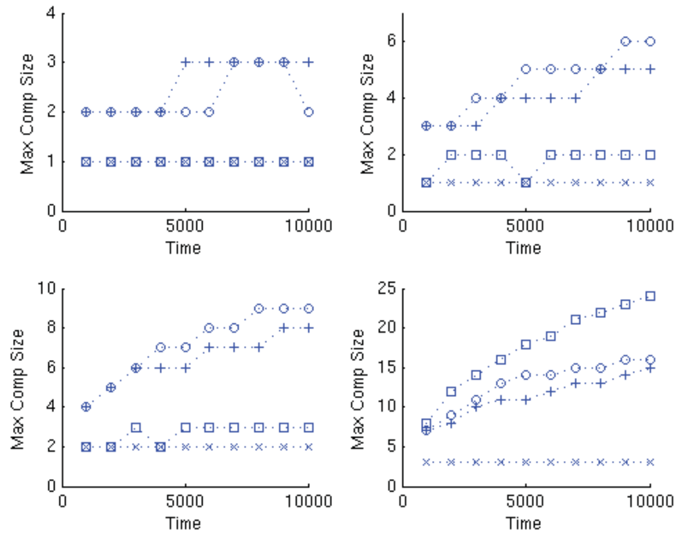


Figure 2. The number of agents in the maximal connected component for simulations in the middle sized environment ($W = 60$). The medians of 100 iterations are plotted. Each panel has a fixed number of agents and varies the jump size. Crosses are $R = 0.5\rho$; circles are $R = 0.6\rho$; squares are $R = 0.8\rho$; and x's mark $R = \rho$. Upper left: $N = 10$. Upper right: $N = 20$. Lower left: $N = 40$. Lower right: $N = 100$.

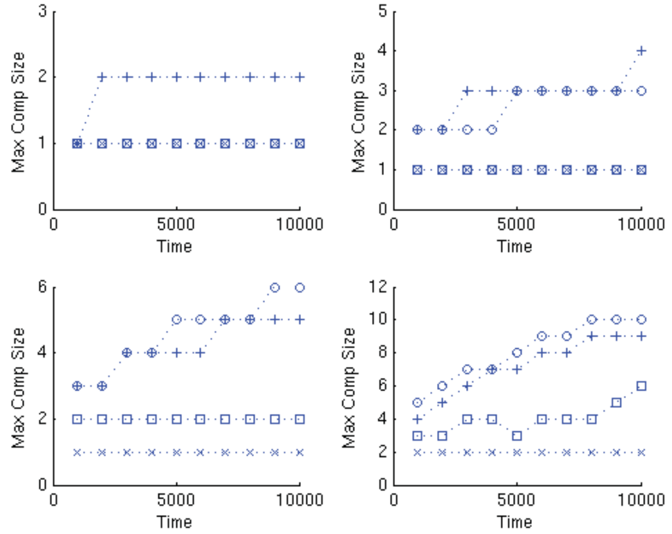


Figure 3. The number of agents in the maximal connected component for simulations in the largest environment ($W = 100$). The medians of 100 iterations are plotted. Each panel has a fixed number of agents and varies the jump size. Crosses are $R = 0.5\rho$; circles are $R = 0.6\rho$; squares are $R = 0.8\rho$; and x's mark $R = \rho$. Upper left: $N = 10$. Upper right: $N = 20$. Lower left: $N = 40$. Lower right: $N = 100$.

A couple of general patterns emerge from the simulations. First, the agents that are initially close to one another quickly aggregate into clusters. Individual agents jump around the domain in steps bounded by R, but the more agents in a cluster, the more the the random jumps are averaged out. So large clusters move around the domain more slowly. Thus, the dynamics act very much like a coalescent process in probability. Small clusters form quickly, but as the clusters get larger their rate of coalescence slows down. At each time slice, the figures plot the number of agents in the maximum connected component in the median simulation run. The largest component then has a fifty percent chance of being bigger (or smaller) than the one plotted.

Second, the density of the agents in the domain is very important to the control. This is not that surprising, but if we compare, say, the lower right panel in the three figures, an interesting pattern is noted regarding the jump sizes. In Fig. 1 (showing the smallest region and thus the most densely packed agents), the larger jump sizes cause more agents to cluster more quickly. In the middle sized region (Fig. 2), the largest sized jump performs the worst, and in Fig. 3, the two largest jump sizes perform the worst. As the jump size increases, the easier it is for clusters of agents to break apart. In the denser regions, large random jumps break clusters apart, but chances are the scattered bits are close to other clusters.

4.2 Normally distributed initial conditions

In this section we perform simulated experiments similar to the previous subsection, except that the initial positions are normally distributed. It is easy to imagine mission scenarios where normally distributed initial conditions are realistic, e.g., agents dropped from a helicopter. Since there will be a greater amount of initial clustering about the mean, we were curious to see if the larger random jumps could force consensus in larger domains (quite unlike the case with uniform initial conditions).

As above, we simulate 100 realizations of the dynamics for 10,000 time steps. Agents are placed in a square 100 by 100 region (the largest considered above) with initial normally distributed initial conditions. The mean of the distribution was set to be the center of the square, and two standard deviations were checked: five and ten times the sensing radius. In none of the simulations was an agent assigned an initial placement outside of the domain. Finally, the two largest jump sizes were simulated: $R = 0.8\rho$ and $R = \rho$. Comparisons with the uniform initial conditions are given in Fig. 4. As before, the median maximum component size of 100 simulation runs is plotted.

5 Planned Simulations and Experiments

For verification of the methods described in this paper, an experimental setup with real robots is employed. The arena for robot exploration is an area that is roughly $5m$ x $3.5m$. The agents in this experiment are Create mobile robots from iRobot. A motion capture video system comprised of twelve OptiTrack V100R2 cameras and the Tracking Tools API from NaturalPoint is utilized to provide an indoor GPS capability. Each Create robot is assigned a unique pattern of reflective markers and a corresponding “rigid body” definition in the software, allowing

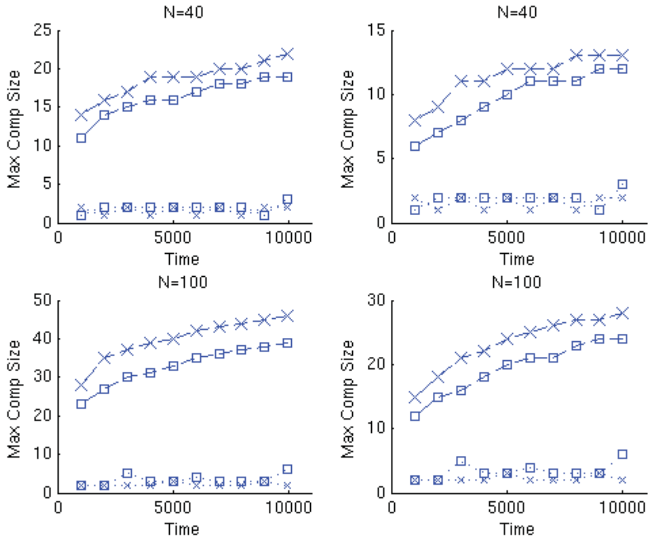


Figure 4. The number of agents in the maximal connected component for simulations in the largest environment ($W = 100$) with normally distributed initial conditions. The medians of 100 iterations are plotted. Each panel has a fixed number of agents and varies the jump size. Normally distributed initial conditions are connected with dashes (upper trajectories) and uniformly distributed initial conditions are connected with dots (lower trajectories, identical to Fig. 3 lower panels). Squares are $R = 0.8\rho$; and x's mark $R = \rho$. Upper left: $N = 40$ and stan. dev. of initial position is $5R$. Upper right: $N = 40$ and stan. dev. of initial position is $10R$. Lower left: $N = 100$ and stan. dev. of initial position is $5R$. Lower right: $N = 100$ and stan. dev. of initial position is $10R$.

reasonably robust tracking of the agents at 100Hz.

These robots are clearly governed by dynamics that are more complex than single-integrator particles. Experiments will be conducted utilizing real on-board cameras and scanning laser rangefinders (see, e.g., [50, 51]) to ascertain the limitations of this controller under these dynamics, as well as to understand the rate of convergence under these considerations. Additionally, the probabilistic functions proposed in this paper will be empirically verified. Specifically, the shapes of the sensing functions in two (or more) dimensions must be addressed.

6 Conclusion

Swarm agreement under certain probabilistic connectivity limitations has been proven to be asymptotically convergent for bounded spaces. A controller has been developed that conforms to the assumptions in the proof, and simulations verify the convergence under such a controller. Planned experiments will additionally validate these results.

7 ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Army Research Office under grant number W911NF-08-1-0106. Also, a version of this paper was presented at the Society for Industrial and Applied Mathematics Dynamical Systems Conference in the Collective Behavior mini-symposium. We owe a debt of thanks to the mini-symposium's organizers, Mason Porter and Erik Bollt and to the participants in the symposium for their many suggestions. Finally, we thank three anonymous reviewers for their comments and suggestions.

REFERENCES

[1] Reynolds, C. W., 1987. "Flocks, herds and schools: A distributed behavioral model". In SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, ACM, pp. 25–34.

[2] Braitenberg, V., 1984. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, Massachusetts.

[3] Matarić, M., 1992. "Minimizing complexity in controlling a mobile robot population". *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'92)*, May.

[4] Parker, L. E., 1994. "Heterogeneous multi-robot cooperation". PhD thesis, Massachusetts Institute of Technology.

[5] Ampatzis, C., Tuci, E., Trianni, V., and Dorigo, M., 2008. "Evolution of signaling in a multi-robot system: Categorization and communication". *Adaptive Behavior,* **16**(1), pp. 5–26.

[6] Arkin, R., Balch, T., and Nitz, E., 1993. "Communication of behavioral state in multi-agent retrieval tasks". In Proceedings of the 1993 IEEE International Conference on Robotics and Automation (ICRA'93), pp. 588–594.

[7] Balch, T., and Arkin, R., 1998. "Behavior-based formation control for multirobot teams". *IEEE Transactions on Robotics and Automation,* **14**(6), Dec., pp. 926–939.

[8] Chaimowicz, L., and Kumar, V., 2007. "Aerial shepherds: Coordination among UAVs and swarms of robots". In *Distributed Autonomous Robotic Systems*. Springer Japan, ch. 6, pp. 243–252.

[9] Holland, O., Woods, J., De Nardi, R., and Clark, A., 2005. "Beyond swarm intelligence: the UltraSwarm". In Proceedings of the IEEE Swarm Intelligence Symposium (SIS2005), pp. 217–224.

[10] Lawton, J., Beard, R., and Young, B., 2003. "A decentralized approach to formation maneuvers". *IEEE Transactions on Robotics and Automation,* **19**(6), Dec., pp. 933–941.

[11] Mondada, F., Gambardella, L., Floreano, D., Nolfi, S., Deneuborg, J.-L., and Dorigo, M., 2005. "The cooperation of swarm-bots: physical interactions in collective robotics". *Robotics and Automation Magazine, IEEE,* **12**(2), Jun., pp. 21–28.

[12] Olfati-Saber, R., 2006. "Flocking for multi-agent dynamic systems: algorithms and theory". *IEEE Transactions on Automatic Control,* **51**(3), March, pp. 401–420.

[13] Tanner, H., Jadbabaie, A., and Pappas, G., 2003. "Stable

flocking of mobile agents, part I: Fixed topology". In Proceedings of the 42nd IEEE Conference on Decision and Control (CDC2003), Vol. 2, pp. 2010–2015.

[14] Tanner, H., Jadbabaie, A., and Pappas, G., 2003. "Stable flocking of mobile agents part II: Dynamic topology". In Proceedings of the 42nd IEEE Conference on Decision and Control (CDC2003), Vol. 2, pp. 2016–2021.

[15] Chen, F., Chen, Z., Liu, Z., and Yuan, Z., 2009. "Control a state-dependent dynamic graph to a pre-specified structure". *Kybernetika,* **45**(5), pp. 801–808.

[16] Bullo, F., Cortés, J., and Martínez, S., 2009. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press. Electronically available at http://coordinationbook.info.

[17] Hayes, A., Martinoli, A., and Goodman, R., 2001. "Swarm robotic odor localization". In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IRAS'01), Vol. 2, pp. 1073–1078.

[18] Kumar, M., Garg, D., and Kumar, V., 2010. "Segregation of heterogeneous units in a swarm of robotic agents". *IEEE Transactions on Automatic Control,* **55**(3), Mar., pp. 743–748.

[19] Yun, X., Alptekin, G., and Albayrak, O., 1997. "Line and circle formation of distributed physical mobile robots". *Journal of Robotic Systems,* **14**(2), pp. 63–76.

[20] Yamaguchi, H., and Burdick, J., 1998. "Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations". *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA'98),* **4**, May, pp. 3573–3580.

[21] Antonelli, G., Arrichiello, F., and Chiaverini, S., 2008. "The entrapment / escorting mission". *IEEE Robotics and Automation Magazine,* **15**(1), Mar., pp. 22–29.

[22] Lewis, M. A., and Tan, K.-H., 1997. "High precision formation control of mobile robots using virtual structures". *Autonomous Robots,* **4**(4), pp. 387–403.

[23] Leonard, N., and Fiorelli, E., 2001. "Virtual leaders, artificial potentials and coordinated control of groups". In Proceedings of the 40th IEEE Conference on Decision and Control, (CDC2001), Vol. 3, pp. 2968–2973.

[24] Olfati-Saber, R., and Murray, R., 2002. "Distributed cooperative control of multiple vehicle formations using structural potential functions". In Proceedings of the 2002 IFAC World Conference.

[25] Olfati-Saber, R., and Murray, R., 2002. "Graph rigidity and distributed formation stabilization of multi-vehicle systems". In Proceedings of the 41st IEEE Conference on Decision and Control (CDC'02), Vol. 3, pp. 2965–2971.

[26] Bruemmer, D., Dudenhoeffer, D., Anderson, M., and McKay, M., 2002. "A robotic swarm for spill finding and perimeter formation". *Spectrum*.

[27] Mai, C.-Y., and Lian, F.-L., 2006. "Analysis of formation control and communication pattern in multi-robot systems". *Proceedings of International Joint Conference SICE-ICASE, 2006*, Oct., pp. 640–645.

[28] Cvetković, D. M., Doob, M., Gutman, I., and Togašev, A., 1988. *Recent Results in the Theory of Graph Spectra*. Elsevier.

[29] Tutte, W. T., 1984. *Graph Theory*. Addison-Wesley.

[30] Gross, J. L., and Tucker, T. W., 1987. *Topological Graph Theory*. Wiley-Interscience.

[31] Prisner, E., 1995. *Graph Dynamics*. Longman Group Limited.

[32] Bollobás, B., 1998. *Modern Graph Theory*. Springer-Verlag.

[33] Godsil, C., and Royle, G., 2001. *Algebraic Graph Theory*. Springer.

[34] Beineke, L. W., Wilson, R. J., and Cameron, P. J., eds., 2004. *Topics in Algebraic Graph Theory*. Cambridge University Press.

[35] Cvetković, D. M., Rowlinson, P., and Simić, S., 2004. *Spectral Generalizations of Line Graphs on Graphs with Least Eigenvalue -2*. Cambridge University Press.

[36] Mesbahi, M., 2003. "State-dependent graphs". In Proceedings of 42nd IEEE Conference on Decision and Control (CDC2003), Vol. 3, pp. 3058–3063.

[37] Mesbahi, M., 2004. "On state-dependent dynamic graphs and their controllability properties". In 43rd IEEE Conference on Decision and Control, 2004, Vol. 3, pp. 2473–2478.

[38] Mesbahi, M., 2005. "On state-dependent dynamic graphs and their controllability properties". *IEEE Transactions on Automatic Control,* **50**(3), pp. 387–392.

[39] Rahmani, A., and Mesbahi, M., 2007. "Pulling the strings on agreement: Anchoring, controllability, and graph automorphisms". In Proceedings of the American Control Conference (ACC'07), pp. 2738–2743.

[40] Rahmani, A., Meng, J., Mesbahi, M., and Egerstedt, M., 2009. "Controllability of multi-agent systems from a graph-theoretic perspective". *SIAM Journal of Optimal Control,* **48**(1), pp. 162–186.

[41] Kim, Y., and Mesbahi, M., 2005. "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian". In Proceedings of the American Control Conference (ACC2005).

[42] Tan, X., 2008. "Swarming control using parallel Gibbs sampling". In Proceedings of the American Control Conference (ACC'08), pp. 3701–3706.

[43] Hatano, Y., Das, A., and Mesbahi, M., 2005. "Agreement in presence of noise: pseudogradients on random geometric networks". In Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference. (CDC-ECC '05), pp. 6382–6387.

[44] Hatano, Y., and Mesbahi, M., 2005. "Agreement over random networks". *IEEE Transactions on Automatic Control,* **50**(11), Nov., pp. 1867–1872.

[45] Mesbahi, M., 2008. "On the factorization, observability, and identification of the agreement protocol". In Proceedings of the 16th Mediterranean Conference on Control and

Automation, pp. 1610–1615.

[46] Zelazo, D., and Mesbahi, M., 2008. “On the observability properties of homogeneous and heterogeneous networked dynamic systems”. In Proceedings of the 47th IEEE Conference on Decision and Control (CDC2008), pp. 2997–3002.

[47] Bertsekas, D., and Tsitsiklis, J., 1989. *Parallel and distributed computation: Numerical Methods*. Prentice Hall.

[48] Moreau, L., 2005. “Stability of multiagent systems with time-dependent communication links”. *IEEE Transactions on Automatic Control,* **50**(2), Feb., pp. 169–182.

[49] Tahbaz-Salehi, A., and Jadbabaie, A., 2010. “Consensus over ergodic stationary graph processes”. *IEEE Transactions on Automatic Control,* **55**(1), Jan., pp. 225–230.

[50] Fricke, G., Zhang, G., Li, W., Caccavale, A., and Garg, D., 2010. “An intelligent sensing network of distributed swarming agents for perimeter detection and surveillance”. In Proceedings of the 3rd ASME Dynamic Systems and Control Conference (DSCC2010).

[51] Zhang, G., Garg, D., and Fricke, G., 2010. “Hazardous spill perimeter detection and monitoring via multiple autonomous mobile robotic agents”. In Proceedings of the 3rd ASME Dynamic Systems and Control Conference (DSCC2010).

A.2.14 Aggregation and Rendezvous in an Unbounded Domain without a Shared Coordinate System

The following paper (on the next 6 pages) was co-authored by Bruce Rogers, Gregory K. Fricke, and Devendra P. Garg, and appeared in the Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference on pages 1437–1442, published and presented in December 2011.

2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Aggregation and Rendezvous in an Unbounded Domain without a Shared Coordinate System

Bruce Rogers, Gregory Fricke, Devendra P. Garg

Abstract—**Many swarm robotics problems focus on the details of robotic dynamics, while ignoring certain other practical issues such as the boundedness of the exploration space, the probability of unsuccessful communication or sensing between agents, knowledge of a common coordinate system, or the initial distribution of robots. A two-phase control is devised which, under appropriate individual bounds, almost surely leads to successful rendezvous. We examine a likely and practical initial distribution, which leads to an exploration protocol wherein an agent is obligated to stay within a certain range of its initial position. Furthermore, the rate of convergence is primarily limited to the velocity limitations of the agents. We solve the aggregation and rendezvous problem with random initial conditions in an arbitrary, unbounded 2D domain without an *a priori* share coordinate system. We examine rates of successful aggregation under certain initial distributions, and rates of convergence under uncertain communication.**

I. Introduction

The aggregation and rendezvous problems are basic hurdles in swarm robotics and distributed control [1]–[3]. In aggregation, the goal is to develop a distributed control that allows an arbitrary number of agents to act together to achieve some task [4], [5]. For the rendezvous problem, the goal is for the agents to converge to a common position. We assume that the agents do not have a shared coordinate system and operate in an unbounded two-dimensional domain. Also, we assume communication constraints between the agents that are realistic for current technology. However, the control developed here is high level and rather abstract; we are not concerned with issues like obstacle avoidance, and the agents are assumed to be point masses operating in discrete time.

In contrast to our assumptions, rendezvous is relatively simple with a shared coordinate system and no bounds on communication. Each agent can broadcast its position to the others; they calculate the average position and move to that point. If communication is only local, averaging algorithms may produce many clusters of agents instead of a single swarm. Without shared coordinates, robots must first sense the position of neighbors to develop a shared map. In this paper, we explore a two stage control protocol, first utilizing random motion that allows the agents to develop shared coordinates. The agents then use local averaging to solve the rendezvous problem under some realistic constraints.

Communication restrictions between agents depend on mission specific parameters: the sensors of the robots, the size of the domain, etc. We make the following assumptions on the communication abilities of the agents and their ability to infer others' positions.

- **No shared map:** Agents must meet or sense each other in order to share coordinates.
- **Sensing radius:** Each robot is equipped with a sensor, such as a camera or laser range finder. Most commonly, such sensors have a sweep of less than 360 degrees, and exhibit sensing errors.
- **Simultaneous communication:** Each agent can maintain at most k simultaneous communications. For example, standard Bluetooth technology allows $k = 7$ simultaneous communications. It's important to note that it is not necessary to keep the same k partners over time.
- **Probabilistic communication radius:** Robots have a limited energy source and may be examining an area much larger than the power of the antennae can cover. In general, the probability of a successful communication decreases with distance. In what follows, we simplify matters in one of two ways: by assuming a fixed probability p_d of the failure of each communication attempt, or by assuming that the probability of a successful communication decays with distance.

We develop a control where agents first move randomly to create a cache of comrades with common coordinates. Each agent then selects some small number (bounded by k) of individuals from its cache and moves to the average position of those individuals. In Section II we give a random motion paradigm that supplies the agents with common coordinates and provides bounds on the time through simulations. In Section III we discuss the global dynamics of the averaging portion of the control. Section IV proposes future simulations and experiments and offers some concluding remarks.

II. Finding other agents in the field

The agents in our system wish to aggregate, but there is one major impediment: They do not have a shared coordinate system. They must sense each other in the field before communicating their positions to each other; it would be meaningless otherwise. Upon sensing each other, each agent records the sensed position of the newly met agent; in the following communication step, the agents share their own beliefs of their current and starting positions within their own coordinate frame. In this way, the coordinate frames of all agents become united. In fact, upon complete successful aggregation, each agent has a complete map of the starting points of the entire agent set relative to its own coordinate frame.

978-1-61284-801-3/11/$26.00 ©2011 IEEE

Let $x_i(t) \in \mathbb{R}^2$ be the position of agent i at time t. If there are N agents, the state space of the system is a subset of $\mathbb{R}^{2\times N}$. In applications, the state vector may include additional coordinates, e.g. velocity or orientation; however, without loss of generality, we treat aggregation as a purely positional problem.

We suppose that each agent has some open, bounded, simply connected sensing domain $D_i(t) \subset \mathbb{R}^2$, and that the shape of the D_i is the same for each i and all times. Since our agents are point masses, we assume without loss of generality that $i \in D_i(t)$ for all i and t. We also suppose the sensors are less than perfect; there is a failure probability p_s at each sensing opportunity. If $x_i(t) \in D_j(t)$ at time t, agent j senses i's presence with probability $1 - p_s$. If $x_i(t) \in D_j(t)$ *and* $x_j(t) \in D_i(t)$ at some time t, then agents i and j exchange coordinate systems and add each other's communication address (e.g., IP address) to their phone book with probability $(1-p_s)^2$. Also, since the agents can remember all agents they have previously encountered, two agents exchange not just their own coordinates upon sensing, but also the entire contents of their address and phone books.

At the abstract level, we need a protocol for agents to wander about and find each other. However, if the environment is unbounded and the agents move randomly (e.g., Brownian or Lévy motion), it's possible some of the agents will wander away from the rest. To prevent that possibility, we give the agents a wandering radius W_r. The agents' motions are restricted so that $\|x_i(0) - x_i(t)\| < W_r$ for all i, which is a simple control to implement since each agent has its own relative coordinate system. Let B_i be the ball of radius W_r centered at $x_i(0)$; this is agent i's wandering domain. Also, let D be the support of the initial distribution of agents, so that $x_i(0) \in D \subset \mathbb{R}^2$ for all i.

There is a natural graph theoretic exposition of the control. For basic graph theory definitions, see [6] for a classical treatment or [7]–[11] for an engineering viewpoint. We consider two different networks. First, at each time t, there is a sensing network $\mathcal{G}_t = (V, E_t)$ where the agents are the set of nodes and i is connected to j if and only if $x_i(t) \in D_j(t)$ and $x_j(t) \in D_i(t)$. Because distant agents may communicate if they have met previously, the union of all previous connections is key to aggregation. Then for a fixed time $T > 0$, set $\mathcal{G}(T) = (V, \bigcup_t^T E_t)$. That is, $i \sim j$ in $\mathcal{G}(T)$ if $i \sim j$ in $\mathcal{G}_t$ for any $t \leq T$. The idea of accumulating a set of changing edges is present in [14] and [12]; however, in those works the topology of the network does not explicitly depend on the geometry of the state space. Here, the cumulative sensing graph $\mathcal{G}(T)$ depends on how the wandering domains B_i overlap since two agents i and j can sense each other only if the distance between B_i and B_j is small enough (and certainly if $B_i \cap B_j \neq \emptyset$).

With this in mind, we next define the network $\mathcal{G}_B$ where the nodes are the sets B_i, and two wandering domains are connected if $B_i \cap B_j \neq \emptyset$. $\mathcal{G}_B$ is sometimes called the *nerve* of the cover $\bigcup B_i$. The following proposition essentially follows from the definitions.

Proposition 1: Suppose the dynamics of the individual agents are given by independent stationary ergodic stochastic processes such that the support of the stationary distribution of $x_i(t)$ is B_i. Then if the network of the wandering domains $\mathcal{G}_B$ is connected, for any $\epsilon > 0$ there is a (random) finite time τ such that the probability the cumulative sensing network $\mathcal{G}(\tau)$ is connected is $1 - \epsilon$.

Proof: It suffices to show that two agents in intersecting regions will sense each other in finite time. This is bound to happen by the recurrence property of ergodicity. Given any disc in the intersection of the regions, there is a finite time where both agents will be in any disc. Make the disc sufficiently small so that the agents must sense each other. If $p_s > 0$, the probability of sensing goes 1 as the two agents recurrently meet in the disc. ■

The "almost surely" in the proposition is with respect to the stationary measure. Also, the wandering domains do not need to be discs, but we use discs below because the agents have random initial orientations. In what follows, each agent i picks a point uniformly at random from B_i and moves there, and then repeats. These dynamics clearly meet the hypotheses of the Proposition. In the next subsection, we show by simulation that connection can be achieved in a reasonable amount of time.

Once $\mathcal{G}(t)$ is connected, the agents have essentially solved the aggregation problem. There is then a chain of contacts between any two agents, and thus any two agents can pass communications through that chain. Then every agent's address book will contain the phone number and coordinate system of every other agent. One issue is to develop a communication protocol so that the agents' address books are built up over time, and to minimize the adverse effects of call drops. Another issue not addressed in this paper is the required self-localization of each agent relative to its own coordinate frame; there are many methods in the literature addressing this practicality through odometry calibration and estimation [15], SLAM methods [16], [17], or (perhaps most relevant to this discussion) relative localization [18], [19].

A. Simulations for aggregation time

In this section we wish to determine how long it will take to connect the network $\mathcal{G}(t)$ with a realistic number of agents deployed in a realistic initial domain D. We focus on a small team of agents, $10 \leq N \leq 20$ with initial positions drawn from a two-dimensional normal distribution. For $N \to \infty$, the connectivity of $\mathcal{G}_B$ is well understood [13], but very large systems of robots are not feasible at this time.

We first need to determine constraints on the time until connectivity in best conditions before considering less ideal sensing. In the simulations, we assume perfect, omnidirectional sensing. That is, each agent has a sensing radius ρ, and two agents sense each other at time t if $\|x_i(t) - x_j(t)\| < \rho$. This represents a best case scenario of the discussion above. Without loss of generality, set the sensing radius of the agents to unity, $\rho \equiv 1$.

Suppose the initial position of each agent is drawn from a normal distribution with mean at the origin. The standard deviation in the x-coordinate is five times the sensing radius, and the standard deviation in the y-coordinate is the same as the sensing radius. (The covariance between x and y is 0). Such a distribution may be realistic in a scenario where the agents are, say, ground robots deployed by an aircraft in flight; a long, thin initial distribution makes sense. Under these chosen "mission parameters", the wandering radius W_r of each agent is set to 14. This value was chosen so that the wandering domain network ($\mathcal{G}_B$ above) is very likely to be connected. In simulations, the wandering domain networks were *not* connected only 16 times out of 11,000 instantiations. Additional experiments were run for $W_r = 10, \ldots, 20$. Clearly, the larger the radius, the more likely $\mathcal{G}_B$ is to be connected, but also the longer it takes for the cumulative sensing network to become connected because each agent has to search a larger area.

With mission parameters defined, we implement a simple agent dynamics. For each $t > 0$, $x_i(t)$ is chosen uniformly at random from agent i's wandering domain. The index t does not represent time in the sense of a ticking clock, but rather the number of actions the agents perform. These dynamics are chosen because we want the agents to aggregate as quickly as possible, but they can sense each other only if they are near enough to one another; independent draws from a uniform distribution mix the agents quickly. Also, for two agents it is easy compute the mean and variance of the sensing time.[1] For more than two agents, there are dependencies that make calculating the expected connection time of $\mathcal{G}(t)$ difficult.

Each agent moves from $x_i(t)$ to $x_i(t+1)$ in our simulation by utilizing a 2^{nd}-order unicycle model, i.e., one that is appropriate for control of a two-wheeled differential-drive mobile robot. At each simulation step, the proximity to other agents is evaluated. If one or more other agents is within the defined sensing range ρ, the agents are said to have met.

Figure 1 shows simulation results for $N = 10, 11, \ldots, 20$. For the initial distribution given above and each value of N, we calculate the first time t when the cumulative sensing network $\mathcal{G}(t)$ is connected. The times t are random, so we run the simulation 1000 iterations for each value of N with fresh initial conditions every iteration. The figure shows the $50^{th}, 75^{th}, 90^{th}$ and 99^{th} percentiles of the simulated aggregation times, as well as the worst case.

The y-axis in Figure 1 measures time in *number of random positions explored*. We see little improvement in performance as N increases from 10 to 20, and the times are very tightly distributed about the median (bottom line). Very rarely, the agents take longer to aggregate; see the crossed line corresponding to the 99^{th} percentile. This happens when the wandering domains have very small overlap. The results are promising: in 90 percent of the 1000 cases, only about 10 positions are necessary for the communication graph to become connected.

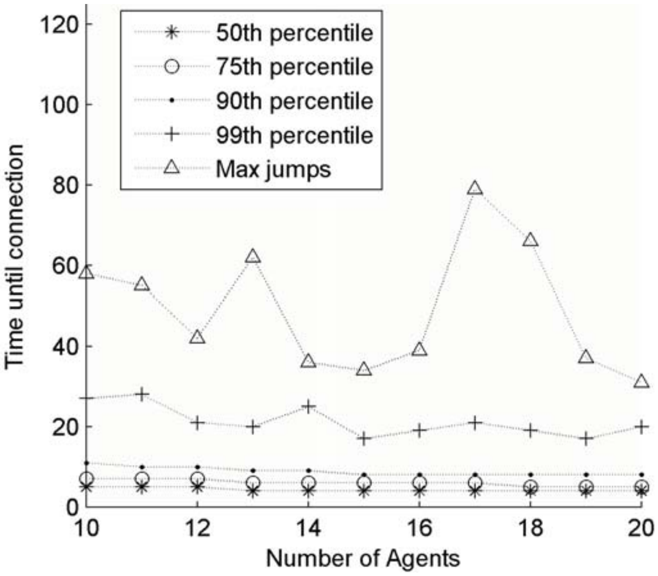


Fig. 1. Time until aggregation for $N = 10$ to 20 agents. The time is random, so 1000 iterations are simulated for each value of N, and percentiles are reported. The time units are number of positions explored.

III. Global dynamics of averaging protocol

We now move to the rendezvous problem with communication constraints. Once the agents have aggregated and can share coordinates, they could broadcast their positions and rendezvous at the mean. However, broadcasting does not necessarily scale with the number of agents, whereas small-scale directed messaging does [20]. If, for example, there are dozens or hundreds of agents, each agent would have to *receive and process* all of those broadcast messages, or there would need to be a protocol to somehow synchronize the messages. Alternatively, if each agent only sends directed messages to a small number of other agents, results from random graph theory confirm that the typical number of incoming requests will be closely related to that small number; thus most agents will have a tenable number of requests to which to respond. As noted in the introduction we assume for practicality reasons that at most k simultaneous communications are possible.

We wish to keep communications to a minimum in order to minimize the detrimental effects of dropped calls. Suppose each agent i nominates exactly two neighbors i_1 and i_2. In this section we consider the global dynamics from the local updating

$$x_i(t+1) = \frac{x_{i_1}(t) + x_{i_2}(t)}{2}. \quad (1)$$

So at each time step, agent i goes to the mean of its two neighbors' previous positions. In the limit as $N \to \infty$, these dynamics are known to contract the unit disc, but the situation is less clear for a fixed finite number of agents.

[1] The sensing time is a Bernoulli random variable with probability proportional to the areas of overlap.

A key practical concern is how the agents choose their two neighbors. The mathematical ideal is for each agent to choose from the others uniformly without replacement. If the agents have aggregated as in the previous section, they have access to the phone numbers of all the other agents and can choose from them uniformly. Even in the ideal case, there is no mathematical guarantee that the positions of the agents will converge to a common value. However, for large N with initial positions randomly distributed in a convex subset of $\mathbb{R}^2$, there are several heuristic reasons to believe the dynamics in (1) will cause the agents to converge. First, the dynamics define a contraction, so the most distant agents are bound to come closer together. Second, since each agent nominates exactly two neighbors, random graph theory tells us that the network of connections between agents will have a unique giant component for large N. That is, the vast majority of agents will be in the same communication class, so under (1) the positions of these agents will converge to the same value.

Finally, even if the communication network has many separate communication classes, the initial conditions are random, so the law of large numbers says that each communication class will converge to a point near the mean of the distribution. These observations fall well short of rigorous proof, but they are played out in simulation. We ran 1000 realizations of the dynamics with $N = 10, \ldots, 20$ and initial conditions uniformly distributed in a rectangle of size $W = 100$ on each side. These initial conditions are more dispersed than the more realistic distribution described in the previous section, so convergence should be less likely and slower. After 25 time steps of the dynamics from (1) we measured whether or not all the agents settled within a disc of diameter 1. If not, we consider this a failure and record the failure rates in the first row of Table I. For the sake of space, we record just the even numbers of agents. The others are similar.

TABLE I
NUMBER OF FAILURES IN 1000 SIM RUNS.

Number of agents	10	12	14	16	18	20
Case 1	29	32	25	36	21	16
Case 2	0	4	8	2	0	3

The second row shows the same measurement, but with a twist on the dynamics. Each agent picks two agents randomly and follows (1) for 10 time steps. Then each agent picks a fresh pair of neighbors and follows (1) for 10 more time steps. In the first case, agents in over 95 percent of the simulations converge to within 1 unit of each other after 25 time steps. In the second case, agents in more than 99 percent of the simulations converge to within 1 unit of each other after just 20 time steps. As in the previous section, time is measured in calculations performed, and the convergence is fast enough that it is practically limited only by the speed of the robot.

Even when the agents do not converge to a single point within 25 times steps (reported as a failure in the table), they come together into several clusters that lie on a common line in the plane. Even in the cases of "failure" the maximum distance between the agents is less than 3. Also, for $N = 250$ or 500 we found that the dynamics converge to within a disc of diameter 1 in every case out of 2000.

A. Probabilistic communication: Fixed probability

We now consider the case where successful communication during the rendezvous phase is not guaranteed. Intuitively, this should decrease the rate of convergence (and perhaps the likelihood of success). To investigate this, we modified the rendezvous simulation such that with probability p_d a communication attempt at a given time-step will be unsuccessful. In this case, a "communication attempt" is the request from one agent to one of its two nominated neighbors for their current positions. An unsuccessful attempt means that the requested information is not returned. If only one such request is successful, the requesting agent will move to the average of its current position and the responding agent's position. If neither request is successful, the requesting agent does not move. (Note that the responding agents do not take this communication into account in generating their own moves.)

We ran three sets of simulations of 100 runs each for values of $p_d = \{0.1, 0.5, 0.9\}$ and $N = 20$, and the results are shown in Figure 2. The y-axis of the figure is the maximum separation between any two agents. The plotted results give the median distances at the given time-step for each of the three values of p_d. Additionally, for the $p_d = 0.9$ case, the 25th and 75th percentile values are shown.

Astoundingly, even with a failure rate of $p_d = 0.5$ (that is, one out of two communication attempts is unsuccessful), the median max-distance is effectively zero before the 30th time-step. Compare this to the case of the previous subsection, i.e., perfect communication, where we mark the rendezvous successful if the max distance is ≤ 1 at completion of the 25th time-step. As expected, the rendezvous requires more time under decreased probability of successful communication; unexpectedly, the effect seems to be quite mild. Even for the case of $p_d = 0.9$, the rendezvous is successful. The rate is clearly much slower in this case, but successful rendezvous can still be expected. These results illustrate a lower bound for the time required for rendezvous under very ineffective communication.

B. Probabilistic communication: Range-dependence

We continue this investigation by making a more realistic assumption regarding the communication: the probability of successful communication decreases with increased range. We know well that the power level of radio frequency signals drops off as the square of the distance, so we might similarly expect that digital signals are less likely to be successfully received at greater distances.

With this motivation, we define the probability of successful communication as a function of the distance d between

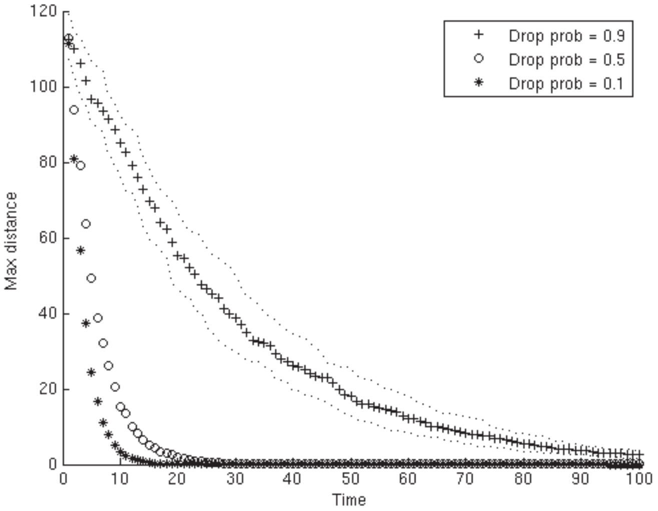


Fig. 2. Convergence rates subjected to various probabilities of successful communication.

any two agents attempting to communicate. For simplicity, we choose the sigmoid function shown in (2), where C_r is the communication radius and a is a shaping parameter that defines the sharpness of the dropoff.

$$p_d(d) = 1 - \frac{1}{1 + e^{a/2 - d*a/C_r}} \tag{2}$$

Choosing $W = 100$ (as previously), $C_r = 0.75W$ and $a = 5$, we have $p_d(0) = 0.924$ and $p_d(\sqrt{2}W) = 0.000979$ (where $\sqrt{2}W$ is the max separation any two agents could experience in this scenario).

We repeated the simulations for convergence using this function for communication probability. In 10,000 iterations, only two instantiations failed to converge within 100 time steps. The results of these simulations are given in Fig. 3 with the crossed points indicating the median max-separation between any two agents, and the dotted lines indicating the 10^{th} and 90^{th} percentiles. From the figure, it is clear that the convergence rates under this assumed function maintain the convergence rates at roughly those of the case where *all* communication is subjected to a 50% probability of failure (regardless of distance). Furthermore the results are very promising in the very tight distribution of convergence rate. It is likely that the shape of this convergence is highly dependent on the shape of the communication function (2), specifically on the sharpness of the cutoff.

IV. Discussion and future directions

We have considered distributed controls that are functions only of an agent's position. Real robots move in continuous time and continuous space, have limitations on their ability to dead-reckon their own positions, and have limitations in their ability to consistently and accurately measure the positions of other agents. In the future we seek to further refine our simulations to address these issues in more realistic and practical terms.

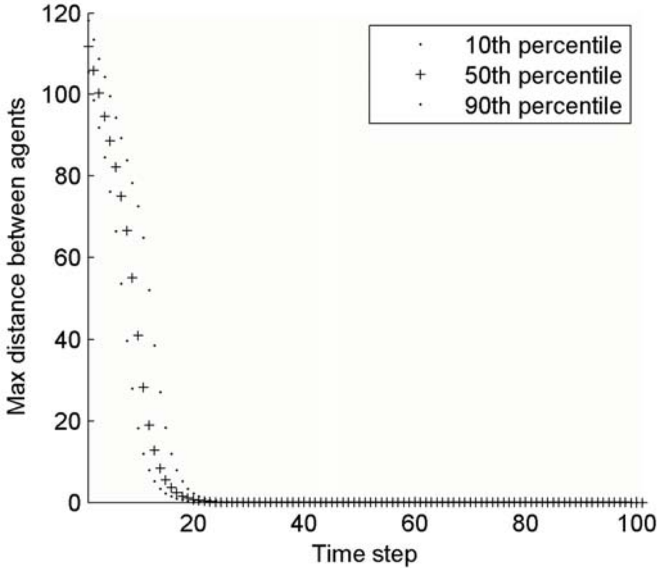


Fig. 3. Convergence rates subjected to the probability of successful communication defined as a function of distance.

A serious challenge is to rendezvous a relatively small number of agents in a perhaps vast environment. We have given a two stage process where agents first find a shared coordinate systems and are able to communicate with each other. Then, each agent moves directly between two other agents in order to rendezvous at a common point. We choose only two others in order to limit dropped communications due to technology and protocol constraints.

We have been silent on how the agents know when to switch from one dynamics protocol to the other. One way is to suppose the agents know how many other agents there are, so each one has an address book with $N - 1$ slots. When an agent's address book becomes full, it votes to switch from the random dynamics of Section II to the deterministic dynamics of Section III. Once the vote is unanimous, the switch is made. This, however, adds requirements to the communication protocol.

A simple alternative method of deciding when to switch to the rendezvous protocol is to capitalize on the results of Section II. Specifically, we see that, for 10 or more agents utilizing this protocol, arbitrarily setting the time-to-switch to, say, $t = 100$ means that with high confidence the address books of all agents will be full. As the number of agents increases within the same distribution (i.e., the initial density increases) the number of time-steps additionally decreases. Clearly then, and especially within the mindset of swarming robotics wherein exactness is of less importance than generalized robustness, a time could be chosen within which some minimum number of agents (with a perhaps larger set of deployed agents) aggregate and rendezvous with a dictated likelihood appropriate for the given mission.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Army Research Office under grant number W911NF-08-1-0106.

REFERENCES

[1] F. Bullo, J. Cortés and S. Martínez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[2] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[3] A. Ganguli, S. Susca, S. Martinez, F. Bullo, and J. Cortes, "On collective motion in sensor networks: sample problems and distributed algorithms", in in *Proceedings of 44th IEEE Conference on Decision and Control and 2005 European Control Conference. (CDC-ECC '05)*, Dec. 2005, pp. 4239–4244.

[4] V. Gazi and K.M. Passino, "Stability analysis of swarms", *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[5] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control", *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1208–1214, 2005.

[6] B. Bollobás, *Modern Graph Theory*. Springer-Verlag, 1998.

[7] M. Mesbahi, "State-dependent graphs," in *Proceedings of 42nd IEEE Conference on Decision and Control (CDC2003)*, vol. 3, Dec. 2003, pp. 3058–3063.

[8] ——, "On state-dependent dynamic graphs and their controllability properties," in *43rd IEEE Conference on Decision and Control, 2004*, vol. 3, Dec. 2004, pp. 2473–2478.

[9] ——, "On state-dependent dynamic graphs and their controllability properties," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 387–392, 2005.

[10] ——, "On the factorization, observability, and identification of the agreement protocol," in *Proceedings of the 16th Mediterranean Conference on Control and Automation*, Jun. 2008, pp. 1610 –1615.

[11] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[12] L. Moreau, "Stability of Multiagent Systems With Time-Dependent Communication Links," *IEEE Transaction on Automatic Control*, vol. 50, no. 2, pp. 169-181, 2005.

[13] M. Penrose, *Random Geometric Graphs*. Oxford University Press, 2003.

[14] A. Tahbaz-Salehi and A. Jadbabaie, "A Necessary and Sufficient Condition for Consensus Over Random Networks, *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 791–795, 2008.

[15] G. Antonelli, S. Chiaverini and G. Fusco, "A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation," *IEEE Transaction on Robotics*, vol. 21, no. 5, pp. 994–1004, 2005.

[16] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005.

[17] A.I. Eliazar and R. Parr, "DP-SLAM 2.0," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA'04)*, vol. 2, Apr. 2004, pp. 1314–1320.

[18] S.I. Roumeliotis and G.A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[19] J. Pugh, X. Raemy, C. Favre, R. Falconi and A. Martinoli, "A Fast Onboard Relative Positioning Module for Multirobot Systems", *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 151–162, 2009.

[20] W. Agassounon, "Distributed information retrieval and dissemination in swarm-based networks of mobile, autonomous agents," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS'03)*, Apr. 2003, pp. 152–159.

A.2.15 Swarm Formations under Nonholonomic and Numerosity Constraints

The following paper (on the next 7 pages) was co-authored by Gregory K. Fricke, Kevin M. Lieberman, and Devendra P. Garg, and appeared in the Proceedings of the 5th ASME Dynamic Systems and Control Conference, published and presented in October 2012.

ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the
JSME 2012 11th Motion and Vibration Conference
DSCC2012-MOVIC2012
October 17-19, 2012, Fort Lauderdale, Florida, USA

DSCC2012-MOVIC2012-8789

SWARM FORMATIONS UNDER NONHOLONOMIC AND NUMEROSITY CONSTRAINTS

Gregory K. Fricke[*]**, Kevin M. Lieberman, Devendra P. Garg**
Dept. of Mechanical Engineering & Materials Science
Duke University
Durham, North Carolina 27708
{gregory.fricke, kevin.lieberman, dpgarg}@duke.edu

ABSTRACT

In this paper, we propose a fully distributed, scalable method of controlling agents with nonholonomic constraints using a Morse potential function. This method successfully controls a swarm of differential-drive (unicycle-type) agents to stable and predictable formations whose structures are not defined *a priori*. The system achieves a stable, minimal energy state.

We consider the effect of *numerosity constraints*, as observed in birds and fish in their shoaling and flocking behavior as a mechanism of reducing complexity, in the interest of achieving fully distributed control over a swarm of any size. The application of numerosity constraints to a swarm system allows the swarm to grow without bound and with no increase in required processing capability of the individual agents. We explore this parameter as a method of minimizing processing and storage requirements while still achieving the qualitative swarm performance. Results from simulations are given for swarms ranging in size over N={6,...,100} acting under our proposed controller as applied to differential-drive (unicycle-type) robots.

1 INTRODUCTION

Distributed control of agents in a swarm is desirable for high robustness in dynamic environments and against individual agent failures. In many applications that employ multiple autonomous agents, the ability of the agents to move in formation is highly desirable, even if the formation structure is not explicitly given. This is certainly true for environmental exploration and coverage applications (see, e.g., [1–4]), or escort applications [5, 6]. Stable formation control can reduce the occurrence of agents getting “lost” or separated from the group. In this paper, we examine the effect of “numerosity constraints” on the dynamic formation behavior of a group of differential-drive ground robots under fully distributed control.

The concept of a “numerosity constraint” [7, 8] is important to consider; evidence from natural systems has shown that both birds [9] and fish [10, 11] exhibit such constraints in their shoaling and flocking behavior. Clearly, in a murmuration of starlings each individual cannot keep track of the entire flock that numbers in the thousands, nor can an individual fish keep track of the entirety of its school. Each individual is thus only aware of a finite number of its nearest neighbors, making its own motive decisions based only on this input. This ability is clearly desirable in a scalable application of multiple mobile robots, allowing each individual to bound its input requirements (and accordingly, its memory and processing requirements as well as its communication or sensing requirements) regardless of the overall size of the swarm.

1.1 Background

Formation control of a group of mobile robotic agents continues to advance. Research in formation control has been widely reported by, among many others, Lewis and Tan [12], Balch and Arkin [13], Desai *et al* [14, 15], Sugar and Kumar [16], Yamaguchi and Burdick [17], Egerstedt and Hu [18], Fierro *et al* [19], Olfati-Saber, Fax and Murray [20–24], Lawton *et al* [25], Tanner *et al* [26, 27], Mai and Lian [28], and Viguria and Howard [29]. In many cases, roboticists look to nature for inspiration (see, e.g., [30–32]). Various approaches are employed in controlling formations, from those that are rigidly defined (i.e., all inter-agent distances and/or angles are specified *a priori*) to those that exploit inter-agent distances or configurations that naturally lead to agreement (i.e., potential functions).

[*] Address all correspondence to this author.

A major advantage of potential function-based approaches is that the individual agents don't require "labeling", i.e., the agents are identity agnostic. This greatly reduces the requisite sensing ability as well as processing and memory requirements, and may remove communication requirements (for control) altogether.

Another significant (and in certain ways more desirable) advantage is robustness in the face of agent failure. If some individual agent in a swarm of N agents is somehow compromised (i.e., disabled, damaged, or outright removed) the remaining agents simply achieve (smoothly) the appropriate minimum-energy state for $(N-1)$ agents.

The resultant structures arise naturally from the minimum-energy framework. This may be disadvantageous in certain cases where one or more agents have differing capabilities, and where certain agents should occupy specific positions within the formation. This limitation can be overcome if those specialized agents use slightly different shaping parameters in their own potential functions.

1.2 Outline

The remainder of this paper is organized as follows. Section 2 defines the problem and the proposed solution. Section 3 describes the simulation-based validation of our proposed method, and presents results with and without numerosity constraints relative to predicted minimum values. Section 4 provides discussion of the limitations, future work, and practical considerations, and in Section 5 the paper is concluded.

2 PROBLEM STATEMENT

Given a set of N agents in an appropriately compact space and initially distributed such that the overall network is connected, identify a distributed controller:

– whose implementation is entirely scalable;
– that results in a stable formation for coverage;
– that is feasible for agents with nonholonomic constraints;
– that exhibits smooth behavior, and;
– that is robust to sensing or communication uncertainty.

The assumption that the swarm of agents is initially connected is an important one. For methods to ensure that agents are deployed and can find each other, the reader is directed to [33, 34].

2.1 Proposed Solution

We propose a distributed controller for a swarm of N differential-drive mobile robots moving under a potential function with a numerosity constraint $\overline{N}$ such that $1 \leq \overline{N} \leq (N-1)$. (The only case where $\overline{N}=1$ is feasible is for $N=2$.) The nearest $\overline{N}$ agents to agent i are defined as the neighbors of agent i, and make up the set $\mathcal{N}_i$.

We start with a Morse potential function as implemented by Chuang *et al* [35] as given in Eq. (1). The vector distance from agent j to agent i is $\vec{d}_{ij}$; $U(i)$ is the total potential energy between agent i and its neighbors $j \in \mathcal{N}_i$, separated by Euclidean distances $d_{ij} = \|\vec{d}_{ij}\|_2$. The attractive and repulsive "strengths" are, respectively, C_a and C_r, and the attractive and repulsive length scales are, respectively, l_a and l_r. These shaping constants are restricted to positive real numbers, and $l_a \neq l_r$.

$$U(i) = \sum_{j\neq i}[-C_a e^{-d_{ij}/l_a} + C_r e^{-d_{ij}/l_r}] \qquad (1)$$

The minimum energy for a pair of agents under this potential function occurs at separation d_{min}, defined in Eq. (2).

$$d_{min} = \frac{l_a l_r}{l_a + l_r} \ln \frac{l_a C_r}{l_r C_a} \qquad (2)$$

The shaping constants are restricted as follows:

$$\begin{aligned} l_a > l_r &\implies l_a C_r > l_r C_a \\ l_a < l_r &\implies l_a C_r < l_r C_a \end{aligned}$$

The controller for agent i uses a typical gradient-descent method for potential function controllers, as shown in Eq. (3), where $\vec{F}_i$ is the vector of forces and is of the same dimensionality as $\vec{d}_{ij}$.

$$\vec{F}_i = -\nabla U(i) \qquad (3)$$

In the simple case of an agent capable of motion and acceleration in any direction, this potential function easily accomplishes a stable tiling of agents. This type of motion is rarely realizable in practice, however, so we must consider a method to re-orient the agents to overcome relevant nonholonomic constraints. The dynamic model we consider in this paper is that of the ubiquitous differential-drive robot. The state of agent i is defined as:

$$X_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \\ v_i \\ \omega_i \end{bmatrix} \qquad (4)$$

where the pair (x_i, y_i) represents the world-coordinates of the agent in the plane, θ_i is the agent's heading with $\theta_i = 0$ corresponding to the $\hat{x}$-axis, and v_i and ω_i are, respectively, the linear and angular speeds.

As the system's potential energy is applied to the agents as force they will accelerate, converting the potential energy into kinetic energy. In order to allow the system to dissipate the energy, c_v and c_ω are included as non-conservative damping terms. The control dynamics are then given by Eq. (5) and Eq. (6), where a_i is the linear acceleration and α_i is the angular acceleration.

$$\dot{X}_i = f(X_i) + g(X_i) \tag{5}$$

$$f(X_i) = \begin{bmatrix} v_i \cos(\theta_i) \\ v_i \sin(\theta_i) \\ \omega_i \\ 0 \\ 0 \end{bmatrix}; \quad g(X_i) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ a_i - v_i c_v \\ \alpha_i - \omega_i c_\omega \end{bmatrix} \tag{6}$$

We modify Eq. (3) in acknowledgment of the lack of force-authority in the direction perpendicular to $\hat{\theta}$ due to the nonholonomic constraint. Instead, we set the control force F_i as the scalar product $F_i = \langle \vec{F}_i, \hat{\theta}_i \rangle$, i.e., the projection of the force vector $\vec{F}_i$ onto the (unit-length) heading of the robot. For the case of agents restricted to the plane, $\vec{F} = [F_x, F_y]^T$, and $F = \|\vec{F}\|_2$. The linear acceleration is, of course, $a_i = F_i/m_i$ where m_i is the mass of agent i; without loss of generality, we set the mass of the agents $m_i = 1$.

$$a_i = F_x(i) \cos\theta_i + F_y(i) \sin\theta_i \tag{7}$$

This projection of force appropriately addresses the nonholonomic constraint inasmuch as it prevents the agent from moving in a manner that is not physically possible. However, the agent must have some ability to change its orientation to move in the appropriate direction to its ideal position relative to its neighbors.

The additional control required for alignment is implemented as a combination of state-averaging with neighbors and an "aligning torque" resulting from the artificial forces acting on the robot. The state-averaging term is akin to torsional springs between agent i and its neighbors, whereas the aligning torque is analogous to that acting on a electric dipole in the presence of an electric field. In this way, we extend the equation of energy to include this angular information, as shown in Eq. (8) with the addition of two terms. The first term reflects the energy of the agent's orientation with respect to the potential field $\theta_{dp}(i) = \arctan(F_{y,i}, F_{x,i}) - \theta_i$, and C_p is the strength of the polar moment (which is used as a tuning parameter). The second term captures the error between the orientation of agent i and the average orientation, $\overline{\theta}_i$, of its neighboring agents j.

$$\begin{aligned} U(i) &= U_d(i) + U_\theta(i) \\ U_d(i) &= \sum_{j \in \mathcal{N}_i} [-C_a e^{-d_{ij}/l_a} + C_r e^{-d_{ij}/l_r}] \\ U_\theta(i) &= C_p \|\vec{F}_i\|_2 (1 - \cos\theta_{dp}(i)) + \frac{1}{2} C_\theta (\overline{\theta}_i - \theta_i)^2 \end{aligned} \tag{8}$$

Correspondingly, the controller in Eq. (3) is modified as follows. First, $\vec{F}(i) = -\nabla U_d(i)$. Another feedback term is added in the form of torque. (As above, the angular acceleration is $\alpha_i = \mathrm{I}_i^{-1} \tau_i$, but we set the inertia I_i equal to identity.) The resultant angular acceleration, α_i, is defined in Eq. (9).

$$\begin{aligned} \alpha_i = & C_p [F_y(i) \cos\theta_i - F_x(i) \sin\theta_i] \\ & + C_\theta (\angle[(\sum_{j \in \mathcal{N}_i} e^{\mathrm{i}\theta_j})/\overline{N}] - \theta_i) \end{aligned} \tag{9}$$

3 SIMULATION OF SYSTEM ENERGIES

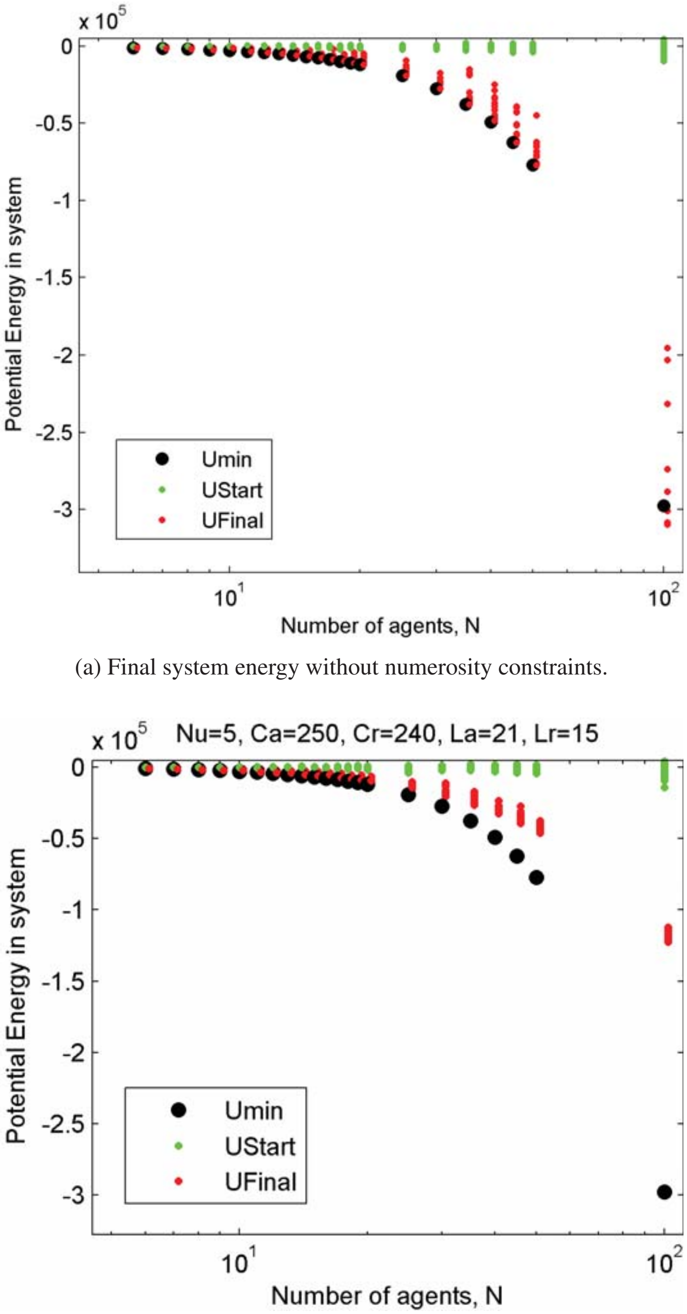
In this section we examine how well the system approaches the minimum energy state. We model teams of agents with sizes of $N = 6, 7, \ldots, 20, 25, \ldots, 50, 100$. The initial positions for each simulation instance are drawn from a two-dimensional uniform distribution, in which the agents are guaranteed to be "close" to one another. (For additional discussion of swarming conditions under which this can be "almost guaranteed" to be connected, see e.g. [33, 34].)

Each agent moves in our simulation under the 2^{nd}-order unicycle dynamic model given above in Eq. (4), i.e., one that is appropriate for control of a two-wheeled differential-drive mobile robot. At each simulation step, the proximity to other agents is evaluated. The neighbor-set for each agent is defined as the nearest $\overline{N}$ agents. The simulations are executed under the continuous-time ODE45 method within MATLAB.

For each value of N, 100 iterations of 500 seconds was run for values of $C_a = 250, C_r = 240, l_a = 21m, l_r = 15m, C_t = 10, C_p = 1, c_v = \sqrt{2}/2, c_\omega = 2$. We compare the results for $\overline{N} = 5$ and $\overline{N} = N$.

Figure 1 shows the energetic results from the simulations described above. The large black dots indicate the minimum achievable potential energy for a planar system of N agents. The small green dots indicate the initial potential energy for each run, and the small red dots (slightly offset to the right for each value of N) indicate the final potential energy.

The minimum achievable energy is found numerically, treating all agents as particles with no motion constraints, but spaced according to the potential function. As this is a numerical approximation, the minimum energy may not be accurate especially for larger N. This is confirmed by the evidence that for a small number of cases, the dynamic simulation achieves a somewhat lower energy than the predicted minimum.



(a) Final system energy without numerosity constraints.

(b) System energy under numerosity constraint of $\overline{N} = 5$.

Figure 1: Final system energy with and without numerosity constraints for swarms of various sizes.

These figures may be somewhat misleading in a few cases, apparently showing that the final potential energy in the system is larger than the initial energy, where the U_{Start} values (in green, and slightly to the left for each case) are higher than the U_{Final} values (in red, and slightly to the right). The reader is reminded that these results are displayed *en mass*; the select cases where U_{Final} values are higher for given N than U_{Start} are from different simulation instances. For every individual case, the final total energy (U_{Final} + kinetic energy) is lower than the initial potential energy.

It is apparent that the systems with numerosity constraints do not achieve the estimated system minimum potential energy. That estimated potential is generated for situations where the swarm agents are each aware of all other agents (i.e., no numerosity constraint). Thus it is true that the numerosity-constrained swarm does not achieve the same value of minimum energy. To explain this, we bring to the reader's attention the X and Y scales on the two figures.

The unconstrained case achieves a significantly denser structure due to the fact that each agent is aware of all other agents; thus, the presence of distant agents actually the swarm spacing. In Figure 2a, the agents are in general separated by $3m$ to $4m$, whereas in Figure 2b the agents are separated by $12m$ to $13m$. In fact, for the parameters used in these simulations, the minimum-energy spacing for two agents by themselves (as calculated from Eq. 2) is approximately $15.5m$. This reveals that by applying numerosity constraints, the typical spacing of agents can be regulated to "near" a design point, d_{min}, regardless of the overall size of the swarm.

Ultimately, Figure 2 illustrates that the overall formation in both cases is more or less properly structured. Additional numerical analysis is underway to fully characterize the error in the formation between the constrained and unconstrained types.

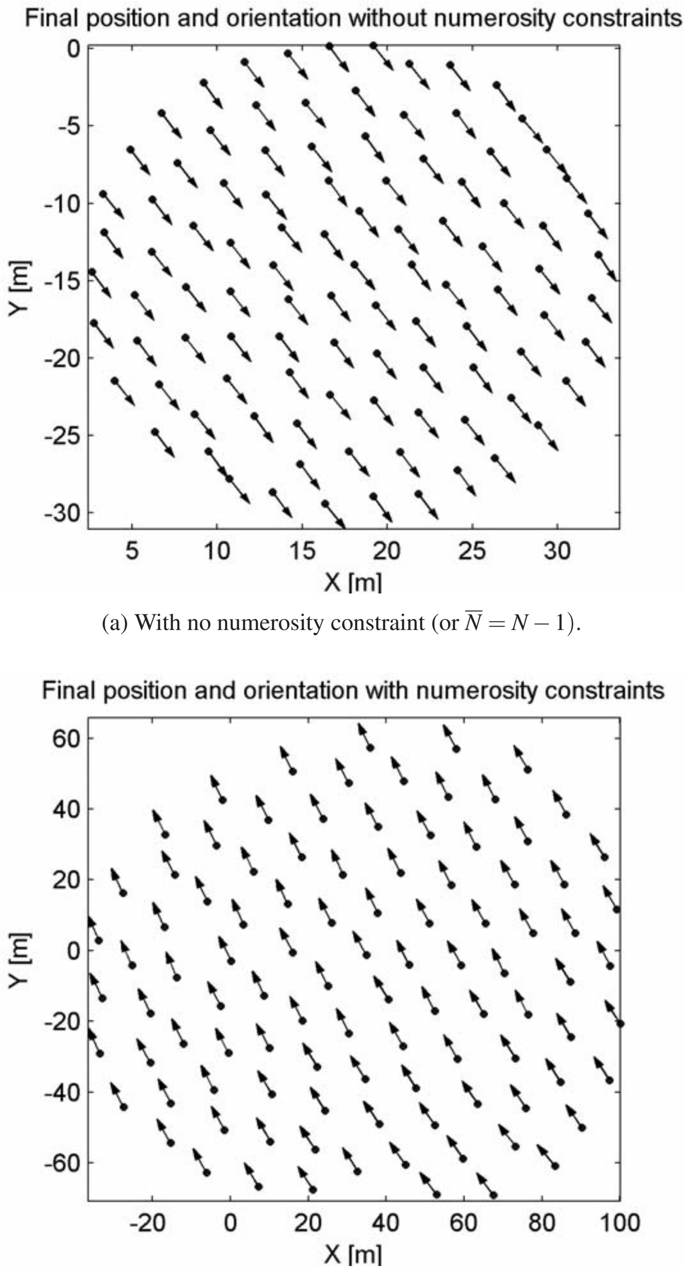
It should be noted that in these figures, the absolute positions and absolute orientations are not critical; this is the very nature of seeking such semi-stable configurations. Only the relative positions and orientations are of importance.

4 DISCUSSION AND FUTURE DIRECTIONS

These simulations assume full, 2π sensing awareness. This is achievable, but not always practical. Future investigation includes limited FOV for the sensors. Additionally, we implicitly assume an infinite sensing range; that is, each agent always has information from its $\overline{N}$ nearest neighbors, regardless of their actual separation. At first glance, this appears to be an egregious assumption; however, in conducting the simulations without this constraint in place, we can empirically observe the separations that actually do occur, and can either tune the potential function parameters such that agents stay within an appropriate (practical) sensing (or communication range), or apply a requirement to real robots such that their sensing range is sufficient to maintain connectivity.

Additional refinement is necessary to determine whether there is a "sweet spot" of numerosity – a particular value, or a ratio of the overall size perhaps. Furthermore the gain values used in the simulation ($C_a, C_r, l_a, l_r, C_t, C_p, c_v, c_\omega$) were selected in a trial-and-error fashion; further tuning of these gains may yield further insights.

In reviewing the dynamic results of the simulations, it is observed that in some cases, one or two agents take large angu-



(a) With no numerosity constraint (or $\overline{N} = N - 1$).

(b) With numerosity constraint $\overline{N} = 5$.

Figure 2: Final positions and orientations of a swarm of 100 agents.

lar velocities – larger than practically feasibly for typical ground robots – indicating that proper the tuning of C_p, C_θ, and c_ω is important.

4.1 Implementation in Practice

We assume communication or sensing constraints between the agents that are realistic for current technology. It is not unreasonable to assume that an individual agent could maintain communication with multiple agents. Bluetooth allows a client-server relationship with up to 6 clients. IEEE802.11 is practically unlimited given one or more access points either in the environment or carried by an agent. This method doesn't even require communication, though; sensing alone can provide the requisite information for this controller.

Ongoing experiments by the authors at the Duke RAMALab have shown that visual detection and position estimation of up to 10 neighbors with simple devices (e.g., off-the-shelf webcams) is achievable at 0.5-1Hz even with the limiting processing power of 600MHz ARM5 processors and only 64MB of system RAM. An alternate implementation on Dell Mini10 Netbooks (1.6GHz Atom processor, 1GB RAM) using the popular OpenCV vision processing package [36, 37] and the extension for color blob-finding [38] implemented in Python is capable of detecting "many" neighboring RAMAbots (augmented with appropriate colors) at up to 11Hz. Such detection and measurement does exhibit some error, which is not yet included in these simulations; further investigation is continuing.

We do not consider the practical consideration of occlusion. In most line-of-sight sensing applications, agents cannot see through one another; thus in the practical sense, the agents are constrained to truly react to their $\overline{N}$ nearest *visible* neighbors. There may be neighboring agents that are physically closer but cannot affect the potential function because it is not sensed. Once the formation structure begins to take shape, though, occlusion should not be an issue if $\overline{N}$ is sufficiently small.

The simulation results presented here do not explicitly place acceleration or speed limits (either angular or linear) on the agents. Continuing work in both simulation and experiment addresses these concerns; these initial results confirm that the controller we propose is feasible with "ideal" agents, and our ongoing effort is focused on such practical considerations.

5 CONCLUSION

We have presented a controller that yields a stable formation for agents with nonholonomic constraints. We additionally allow a constraint to be placed on the agents restricting their awareness of the bulk of the swarm in order to allow scalability, and we have shown that this constraint numerically reduces the ability of the swarm to achieve its minimum possible energy; even under this constraint, however, the swarm is still able to significantly reduce its energy relative to the starting energy, and successfully achieves formation control in a predictable way.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Army Research Office under grant number W911NF-08-1-0106.

REFERENCES

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[2] S. Gabriele and P. Giamberardino, "Redundant coverage for noise reduction in dynamic sensor networks," *WTOS*, vol. 7, pp. 855–865, October 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1486663.1486666

[3] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE Conference on Decision and Control (CDC 2008)*, Dec. 2008, pp. 3947–3952.

[4] F. Zhang, Y. Xi, Z. Lin, and W. Chen, "Constrained motion model of mobile robots and its applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 3, pp. 773–787, Jun. 2009.

[5] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment / escorting mission," *IEEE Robotics and Automation Magazine*, vol. 15, no. 1, pp. 22–29, Mar. 2008.

[6] K. M. Lieberman, G. K. Fricke, and D. P. Garg, "Decentralized control of multi-agent escort formation via Morse potential function," in *Proceedings of the 2012 ASME Dynamic Systems and Control Conference (DSCC2012)*, Ft. Lauderdale, FL, Oct. 2012, to Appear.

[7] N. Abaid and M. Porfiri, "Collective behavior of fish shoals in one-dimensional annular domains," in *Proceedings of the American Control Conference (ACC2010)*, Jun./Jul. 2010, pp. 63–68.

[8] ——, "Consensus over numerosity-constrained random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 649–654, Mar. 2011.

[9] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic, "Empirical investigation of starling flocks: a benchmark study in collective animal behaviour," *Animal Behaviour*, vol. 76, no. 1, pp. 201 – 215, 2008.

[10] R. W. Tegeder and J. Krause, "Density dependence and numerosity in fright stimulated aggregation behaviour of shoaling fish," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 350, no. 1334, pp. 381–390, 1995.

[11] C. Agrillo, M. Dadda, G. Serena, and A. Bisazza, "Do fish count? Spontaneous discrimination of quantity in female mosquitofish," *Animal Cognition*, vol. 11, pp. 495–503, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10071-008-0140-9

[12] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[13] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[14] J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA'98)*, vol. 4, May 1998, pp. 2864–2869.

[15] ——, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, Dec 2001.

[16] T. Sugar and V. Kumar, "Decentralized control of cooperating mobile manipulators," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA'98)*, vol. 4, May 1998, pp. 2916–2921.

[17] H. Yamaguchi and J. Burdick, "Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA'98)*, vol. 4, pp. 3573–3580, May 1998.

[18] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA2001)*, vol. 4, 2001, pp. 3961–3966.

[19] R. Fierro, A. Das, J. Spletzer, J. Esposito *et al.*, "A framework and architecture for multi-robot coordination," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 977–995, 2002.

[20] R. Olfati-Saber and R. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *Proceedings of the 2002 IFAC World Conference*, 2002.

[21] ——, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proceedings of the 41st IEEE Conference on Decision and Control (CDC'02)*, vol. 3, Dec. 2002, pp. 2965–2971.

[22] ——, "Distributed structural stabilization and tracking for formations of dynamic multi-agents," in *Proceedings of the 41st IEEE Conference on Decision and Control (CDC'02)*, vol. 1, Dec. 2002, pp. 209–215.

[23] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sept. 2004.

[24] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[25] J. Lawton, R. Beard, and B. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[26] H. Tanner, G. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, Jun. 2004.

[27] H. G. Tanner and A. Kumar, "Formation stabilization of multiple agents using decentralized navigation functions," in *Robotics: Science and Systems I*. Boston: roboticsproceedings.org, Jun. 2005.

[28] C.-Y. Mai and F.-L. Lian, "Analysis of formation control

and communication pattern in multi-robot systems," *Proceedings of International Joint Conference SICE-ICASE, 2006*, pp. 640–645, Oct. 2006.

[29] A. Viguria and A. Howard, "An integrated approach for achieving multirobot task formations," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 176–186, April 2009.

[30] V. Gazi and K. Passino, "Stability analysis of swarms," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 692 – 697, Apr. 2003.

[31] D. Milutinović and P. Lima, *Cells and Robots: Modeling and Control of Large-size Agent Populations*. Springer, 2007.

[32] G. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control*. Cambridge: MIT Press, 2005.

[33] G. Fricke, B. Rogers, and D. Garg, "On the stability of swarm consensus under noisy control," in *Proceedings of the 4th ASME Dynamic Systems and Control Conference (DSCC2011)*, Arlington, VA, Oct./Nov. 2011.

[34] B. W. Rogers, G. K. Fricke, and D. P. Garg, "Aggregation and rendezvous in an unbounded domain without a shared coordinate system," in *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC2011)*, Orlando, FL, 12-15 Dec. 2011.

[35] Y.-L. Chuang, Y. R. Huang, M. R. D'Orsogna, and A. L. Bertozzi, "Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials," in *Proc. 2007 IEEE International Conference on Robotics and Automation (ICRA'07)*.

[36] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[37] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly Media Inc., 2008. [Online]. Available: http://oreilly.com/catalog/9780596516130

[38] C. C. Liñán, "cvBlob," http://cvblob.googlecode.com. [Online]. Available: http://cvblob.googlecode.com

A.2.16 Decentralized Control of Multi-Agent Escort Formation via Morse Potential Function

The following paper (on the next 8 pages) was co-authored by Kevin M. Lieberman, Gregory K. Fricke, and Devendra P. Garg, and appeared in the Proceedings of the 5th ASME Dynamic Systems and Control Conference, published and presented in October 2012.

**ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the
JSME 2012 11th Motion and Vibration Conference
DSCC2012-MOVIC2012
October 17-19, 2012, Fort Lauderdale, Florida, USA**

DSCC2012-MOVIC2012-8777

DECENTRALIZED CONTROL OF MULTI-AGENT ESCORT FORMATION VIA MORSE POTENTIAL FUNCTION

Kevin M. Lieberman, Gregory K. Fricke, Devendra P. Garg
Dept. of Mechanical Engineering & Material Science
Duke University
Durham, NC 27708
{kevin.lieberman, gregory.fricke, dpgarg}@duke.edu

ABSTRACT

This paper presents a decentralized control algorithm for the Entrapment/Escorting Mission based on the Morse potential function. The entrapment/escorting mission tasks a multi-agent system with creating a formation around a mobile target. Applications of the mission include security awareness and military situations where an autonomous object of value must be guarded. The simulation consists of N robots that are initially located in random positions on a flat terrain. Agents are provided with the information on coordinates of their peers by a supervisor, which are used to generate artificial potential energies between robots and the target. The simulation confirms that robot positions achieve the lowest calculated Morse potential energy in the arena, and the resulting formation is centered on the target. Differential drive "e-puck" robots are used to experimentally validate the control's application to nonholonomic vehicles, and proximity sensors are incorporated into the algorithm to accommodate for obstacles in the experimental arena.

INTRODUCTION

Swarm robotics research is based on a collection of robots, rather than a single agent, accomplishing a specified task. Multi-agent systems offer better dexterity through more controllable components than a single agent, and the effects of failure can be minimized since there are multiple agents redundantly working to complete the same task. While traditional robotics has emphasized dull, dirty, and dangerous applications [1], swarm robotics offers a variety of applications that range from services to life saving operations. Commercial applications include autonomous robots that move goods across warehouse distribution centers [2], and ongoing research challenges a swarm of robots to detect mines in a minefield [3].

Scalability and hardware limitations are obstacles to implementing a multi-agent system. A robust system must be scalable to accommodate for the addition, withdrawal, and failure of robots. However, inter-agent communication can be complex, especially when communication is not centralized or the system includes heterogeneous robot configurations [4]. Swarm robots that are small and depend on batteries will also have relatively little processing power and memory, restricting their abilities.

A major distinction in multi-agent systems can be found between centralized control and decentralized control systems. Centralized control entails communication and direction to come from a "centralized" controller, whereas decentralized control means that each robot is independently controlled. As explained in Ref. [5], centralized control has the benefit of being able to monitor the entire system. Since processing is performed off board, less processing equipment is required on the agent and the cost of each individual agent is decreased. However, centralized control is hindered by high communication requirements and limited processing power. The number of agents the system can accommodate is finite as it is restricted to the controller's hardware limitations. Unlike centralized systems where the entire system risks failure should the central controller fail, decentralized systems equip each robot with the sensors and actuators to independently assert control over itself. Consequently, one agent's failure does not cause the failure of the whole system. The overall performance may be decreased, but the task can still be achieved. More robots can be used since the number of agents is not limited by the capabilities of the central controller. Trade-offs are made as independent processing results in increased costs and complexity.

The emphasis of this research project is on the entrapment/escorting mission [6]. In this mission, multiple robots surround an identified target and are tasked with minimizing access or escapement paths between the target and intruders. Entrapment entails preventing the target from escaping the group of agents; escorting conversely demands that the target is guarded from intruders. The motion of the target is not known *a priori* by the guarding agents. Solutions to the mission have been primarily offered through centralized control, such as through null-space-based behavioral (NSB) control [6], [7], or cluster space control [8]. Reference [9] presents a multi-layer control scheme for centralized formation control, while generating fictitious forces to accommodate for obstacles. As will be discussed, experimental robot limitations in this research violated the assumption in Ref. [9] that agents had sensors that could measure the distance to obstacles. Therefore, a simplified fictitious force calculation is developed in this paper to accommodate for the limitations of the experimental robots.

Decentralized solutions have focused on overall system aggregation. Reference [10] provides decentralized control via algebraic graph theory to a variation of the entrapment/escorting mission. In the modified version of the entrapment/escorting mission referred to as the "surrounding control problem," agents seek to contain an additional set of stationary robots. The focus of Ref. [10] is to design a control algorithm based on a decentralized estimation of the geometric center. In another modified scenario, Ref. [11] challenges a miniature helicopter to follow a group of unmanned ground vehicles (UGV's). While the UGV's moved according to a centralized control algorithm presented in Ref. [9], the helicopter's onboard artificial vision system simultaneously calculates the centroid of the UGV's to guide the helicopter's trajectory. The overall system is decentralized because the UGV and helicopter controllers work independently. The focus of the research presented in this paper is on the control of each agent, rather than the details of overall system aggregation as found in Ref. [10] and [11]. Centroid calculations and estimation techniques are not considered, and the target's motion is dynamic and unknown.

Assumptions made about the mission in this paper differ from the assumptions made in past research. While Ref. [12] examines agent formation around a target, it focuses on how an agent can identify the target and communicate with the rest of the system to inform their formation. This paper, unlike Ref. [12], assumes that all agents are connected, and is therefore not as concerned about agent communication. Additionally, this research does not make the assumption that agents cannot fail. Reference [13] provides decentralized control using grasp theory to constrain a target. Agents move together in a chain in pursuit of the target before enclosing it. However, nonholonomic constraints are not considered, as they are here.

This paper proposes, simulates, and experimentally validates the use of the Morse potential function to provide decentralized control to complete the entrapment/escorting mission. It should be noted that the control algorithm is specifically based on *the* Morse potential (that models the potential energy of a diatomic molecule), rather than the more general Morse *class* of navigation potential functions.

Control objectives include surrounding the target, and minimizing gaps through evenly spacing agents around the target. Obstacle avoidance via onboard proximity sensors is incorporated into the algorithm. The algorithm is scalable and accommodates failure. This methodology builds on the work of Ref. [14] and Ref. [15], which used quadratic artificial potential functions to simulate formations in an arena with assets, threats, and nodes (bodyguard agents). Furthermore, this research experimentally implements decentralized control for nonholonomic vehicles with obstacle avoidance.

The entrapment/escorting mission has a great deal of value for military and security applications that require situational awareness. Given that the Department of Defense has mandated that one-third of all ground combat vehicles be unmanned by 2015 [1], there remain ethical questions of the use of autonomous robots in a war zone. The entrapment/escorting mission replaces guarding tasks typically controlled by humans with autonomous controllers, and the question remains whether robots or remote human controllers can arrive at the same decision as an on-site human in all situations.

BACKGROUND

Multi-robot control frequently draws on biological systems to inspire algorithms for coordination and formation [16, 17]. Potential energy functions have also inspired decentralized control systems by reducing robot behavior to that of particles [18]. By assigning potential energies between robots in the arena, agents naturally move to points of equilibrium at localized energy minima. Scaling potential coefficients can easily change the size of a swarm and offset the equilibrium points. Since potential energy equations are continuous, control is smooth and does not involve toggling between on and off.

This research uses the Morse potential [18, 19] to provide control. The Morse potential represents the attractive and repulsive interaction between two objects, originally used to model the spacing between nuclei in a diatomic molecule. Applications of the Morse potential function to model behavior are quite varied, including modeling crowd dynamics in emergency situations [20]. The expression for the Morse potential is shown in Eq. 1. In the planar case, $x_i = \begin{bmatrix} x \\ y \end{bmatrix}$, whereas $x_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ in 3-D case. In this work, the planar case is considered.

$$U(x_i) = \sum_{j \neq i} \left(C_r e^{-\|x_i - x_j\|/lr} - C_a e^{-\|x_i - x_j\|/la} \right) \quad (1)$$

The scalar parameters *la* and *lr* represent the attractive and repulsive characteristic lengths, and *Ca* and *Cr* are the respective coefficients that scale the potential. The characteristics lengths determine the spacing between agents upon reaching equilibrium. The position of agent *i* is represented by x_i, whereas x_j represents the position(s) of neighbor(s) of agent *i*.

As shown in Eq. 2, the negative gradient of the Morse potential energy is the force exerted on each agent.

$$\vec{F}(x_i) = -\nabla \mathrm{U}(x_i) \quad (2)$$

When two agents are too close to each other, they experience a repulsive force as dictated by the gradient that pulls them away from each other. When the agents are far apart, the force causes them to accelerate towards each other. This attractive force causes agents that are dispersed throughout the arena to sense an attractive force to each other, resulting in a gathering movement. The repulsive force prevents robots from colliding into each other or obstacles. Agents cease their acceleration and settle into the formation when the sum of the repulsive and attractive energies comes to a minimum, resulting in a gradient (and net force) of zero.

The Morse potential function is only dependent on an agent's location and the location of its peers. By providing an agent with this positioning information, each robot can individually calculate the sum of the artificial forces it experiences. This net force then dictates robot acceleration and movement. Assuming each agent can obtain the relevant position information for the system, agent formation and decentralized control can be driven by the Morse potential function. No agent-to-agent communication is required, and the only communication necessary is the positional information. Alternatively, this positional information can be acquired through sensing, eliminating communication requirements altogether. Furthermore, the system is scalable. Agents added to the system (or removed from it) are summed with (or subtracted from) the system energy.

PROBLEM FORMULATION

This research seeks to experimentally validate decentralized control for the entrapment/escorting mission using the Morse potential. As dictated by the mission, agents must surround a target without colliding into each other or the target itself. Additionally, the agents must evenly space themselves around the target. The terrain is assumed to be a flat arena with no holes.

The control algorithm also accommodates for obstacles identified via onboard proximity sensors. Glossy white boxes are randomly placed in the arena, challenging agents to navigate around the obstacles. In the experiments, obstacle positions are neither recorded nor shared between agents.

An agent with randomized motion was designated as the target for the entrapment/escorting mission experiment. Past research has used a tennis ball as a target [6] or a target that emits signals received by agents within a sensing radius [18]. Reference [14] identifies various methodologies for determining a target's motion, such as "randomized waypoints" that creates a random set of destinations for the robot, or "random direction" that randomly assigns a heading for the robot every time it reaches the arena's boundary. In this experiment, the target's controller randomly generates coordinates for waypoints. Upon arrival at a waypoint, the target generates its next waypoint, drawing from a uniform distribution over the traversable area. It is assumed that the target would not attempt to navigate to an unreachable destination. Therefore, the situation when the randomly generated waypoint is coincident with an obstacle is not considered.

The primary assumption in this experiment is that agents know the positioning information of all robots in the system (including the target). In practical applications, this information could be relayed by a GPS or satellite system. Yet it should be noted that there is not a central controller governing the agents' movement. All agents in the simulation and experiment are homogeneous, nonholonomic vehicles.

In the experiments, there was a single target agent in the arena accompanied by four additional agents responsible for surrounding the target. The control method was simulated and experimentally validated with e-puck robots. The e-puck robots were created by the École Polytechnique Fédérale de Lausanne (EPFL) [21], and were selected for their simple communication capabilities via Bluetooth and low cost. The e-puck uses differential-drive with a stepper motor to control each wheel, and thus has nonholonomic constraints. Communication between e-pucks is not possible without extensive customization, reinforcing the need for a decentralized solution to the entrapment/escorting mission. The eight proximity sensors distributed around the perimeter of each e-puck are not accurate enough to provide reliable distance readings across a large range of distances [5], limiting its functionality. However, the proximity sensors can indicate whether an obstacle is immediately present in front of a sensor. As a consequence of this behavior, all collision avoidance in the model utilized the proximity sensors as a detection switch rather than a detection measurement.

ANALYTICAL MODELING

The decentralized controller modeled in this paper features primary movement determined by the Morse potential function. The gradient of the Morse potential in Eq. 1 leads to the modeled force and acceleration that are exerted on the agent. The attractive and repulsive forces between the target and each agent's peers determine the motion of the robot. Since a formation is desired around the target, an additional attractive force towards the target is added to each agent [19]. This results in a total potential energy as seen in Eq. 3.

$$U(x_i) = \frac{1}{2} C_c (x_i - y)^2 + \\ \sum_{j=1}^{N} \left(C_r e^{-\|x_i - x_j\|/lr} - C_a e^{-\|x_i - x_j\|/la} \right) \tag{3}$$

The magnitude of the attractive potential energy to the target is tunable with scalar C_c, x_t is the position of the target, and C_r and C_a scale the respective repulsive and attractive forces between peer agents.

As was shown in Eq. 2, the gradient of the Morse potential can be taken to find the modeled force that the agent experiences in the x and y directions. This results in Eq. 4, where $dx = \left(x_j - x_i\right)$, and $dr = \sqrt{(dx)^2 + (dy)^2}$.

$$\mathrm{m} \frac{dv_i}{dt} = -\nabla U(x_i)$$

$$= \left(C_c (x_i - x_t) + \sum_{j \neq i} \left(\frac{C_a dx}{(la)(dr) e^{\left(\frac{dr}{la}\right)}} - \frac{C_r dx}{(lr)(dr) e^{\left(\frac{dr}{lr}\right)}} \right) \right) \hat{\imath}$$

$$+ \left(C_c (x_i - x_t) + \sum_{j \neq i} \frac{C_a dx}{(la)(dr) e^{\left(\frac{dr}{la}\right)}} - \frac{C_r dx}{(lr)(dr) e^{\left(\frac{dr}{lr}\right)}} \right) \hat{\jmath} \tag{4}$$

This equation verifies that the agent will find equilibrium when the potential energy U is at a minimum. Since the minimum of potential energy is found when its gradient is equal to zero, the agent would not accelerate when the potential energy is at a local minimum.

While this potential function does not consider obstacles impeding agent motion, onboard proximity sensors provide additional guidance. Since the quality of the e-puck proximity sensors was found to not reliably indicate the distance of perceived obstacles, the proximity sensors acted as a switch to indicate whether an obstacle was immediately present. A threshold value was determined, and when a sensor value surpassed the threshold, a repulsive force was applied. When the agent moved away from the obstacle and proximity sensor values dropped below the threshold, the agent returned to the potential function-based controller.

The obstacle repulsive force generated was in the opposite direction of the obstacle. However, a slight offset was subtracted from the vertical away from the obstacle. This offset force allowed the agent to move away from the obstacle while simultaneously advancing around the obstacles perimeter. In Eq. 5, the e-puck generates a matrix indicating which proximity sensor values surpass the threshold.

$$ps_k = \begin{cases} 0 \; if \; k^{th} \; proximity \; sensor < threshold \\ 1 \; if \; k^{th} \; proximity \; sensor > threshold \end{cases} \tag{5}$$

If any of the proximity sensors are above the threshold value, the controller generates an obstacle repulsive force utilizing a matrix that transforms the angle of each proximity sensor to the global reference frame. This is seen in Eq. 6 and Eq. 7. Since the proposed method is designed for an e-puck robot, the angles of the proximity sensors locally on the e-puck are included in the equations. C_{obs} is a coefficient that scales the magnitude of the repulsive force.

$$F_{obs,x,i} = C_{obs} \begin{bmatrix} ps_1 \\ ps_2 \\ ps_3 \\ ps_4 \\ ps_5 \\ ps_6 \\ ps_7 \\ ps_8 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_i - 10^0 + offset) \\ \cos(\theta_i - 45^0 + offset) \\ \cos(\theta_i - 90^0 + offset) \\ \cos(\theta_i - 150^0 + offset) \\ \cos(\theta_i - 210^0 - offset) \\ \cos(\theta_i - 270^0 - offset) \\ \cos(\theta_i - 315^0 - offset) \\ \cos(\theta_i - 350^0 - offset) \end{bmatrix} \tag{6}$$

$$F_{obs,y,i} = C_{obs} \begin{bmatrix} ps_1 \\ ps_2 \\ ps_3 \\ ps_4 \\ ps_5 \\ ps_6 \\ ps_7 \\ ps_8 \end{bmatrix} \cdot \begin{bmatrix} \sin(\theta_i - 10^0 + offset) \\ \sin(\theta_i - 45^0 + offset) \\ \sin(\theta_i - 90^0 + offset) \\ \sin(\theta_i - 150^0 + offset) \\ \sin(\theta_i - 210^0 - offset) \\ \sin(\theta_i - 270^0 - offset) \\ \sin(\theta_i - 315^0 - offset) \\ \sin(\theta_i - 350^0 - offset) \end{bmatrix} \tag{7}$$

Equation 8 subsequently determines the desired velocity and position of the *i-th* agent based on the generated forces, where Eq. 9 defines the velocity of that agent from the previous time step.

$$v = v_i + \Delta v_i \frac{\Delta t}{m} \tag{8}$$

$$v_i = \frac{\Delta x_i}{\Delta t} \tag{9}$$

The modeled mass, *m*, is used as a gain for control purposes, and Δt is the discrete time to complete a loop.

Since the agents used in this research are nonholonomic, the agent's desired velocity must be converted into individual speeds for each wheel. The velocity vector is first converted into polar form in Eq. 10 and Eq. 11.

$$v_{norm,i} = \sqrt{v_{x,i}^2 + v_{y,i}^2} \tag{10}$$

$$\theta_{\mathrm{desired,i}} = \tan^{-1}(\mathrm{V}_{\mathrm{y,i}} / \mathrm{V}_{\mathrm{x,i}}) \tag{11}$$

The desired heading ($\theta_{desired,i}$) is the inverse tangent of the two components of the desired velocity. The heading error is the difference between the desired heading and the agent's current heading (θ_i).

$$\theta_{error,i} = \theta_{desired,i} - \theta_i \qquad (12)$$

Angular velocity can be computed using Eq. 13, where k is a gain.

$$\Omega_i = k * (\theta_{error,i} / \Delta t) \qquad (13)$$

In the model, the gain was scaled so it would be proportional to the magnitude of the artificial force and the heading error.

Upon calculating the magnitude of the velocity and the angular velocity, the values are multiplied by a transformation matrix that provides the linear velocity for each wheel in Eq. 14.

$$\begin{bmatrix} (\omega_{right}) \\ (\omega_{left}) \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & -b/2 \\ 1 & b/2 \end{bmatrix}\begin{bmatrix} v_{norm} \\ \Omega_i \end{bmatrix} \qquad (14)$$

The product of the angular velocity of each wheel, ω, is determined by the radius of the wheel, *r;* the distance between the wheels, *b*, and the solutions to Eq. 10 and Eq. 13.

SIMULATION

The described model was simulated using a commercially available package Webots, marketed by Cyberbotics Ltd. Three simulations were performed with five e-pucks in the arena. One of the e-pucks was designated as the target. The first instance simulated an arena devoid of obstacles, and the second simulation featured two obstacles placed in the middle of the arena as shown in Fig. 1. The third simulation mimicked the first simulation with no obstacles; however, one agent was programmed to temporarily "fail" for 20 seconds to demonstrate the robustness of the system. The agent ceased movement as the other agents continued to move without being disturbed by the failed robot.

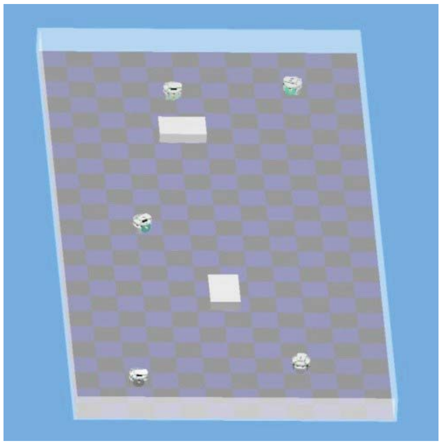


FIGURE 1. SIMULATED ARENA WITH OBSTACLES

In each simulation, agents settled on a formation as shown in Fig 2. Upon creation of the formation, agents remained equidistantly spaced around the target, as shown in Fig. 3. As noted by the arrows in the figure, there was a momentary shift in formation when the target switched direction, causing each agent to change their orientation accordingly. Equidistant spacing was simultaneously achieved when the agents reached their minimum Morse potential energy, as noted in Fig. 4. The agents asymptotically approached energy levels of 0.3 x 10^{-6} J, and maintained a distance of approximately 20cm between the centroid of each agent. Figure 5 depicts the potential energy of an agent at points across the arena. The actual position of the agent marked in white shows that it is located at its lowest potential energy in the arena while in the formation.

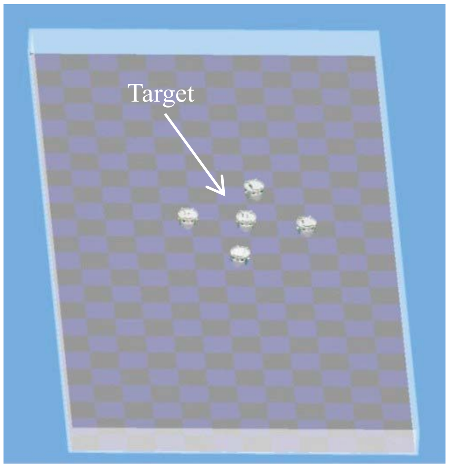


FIGURE 2. SIMULATED AGENT EQUILIBIRIUM FORMATION

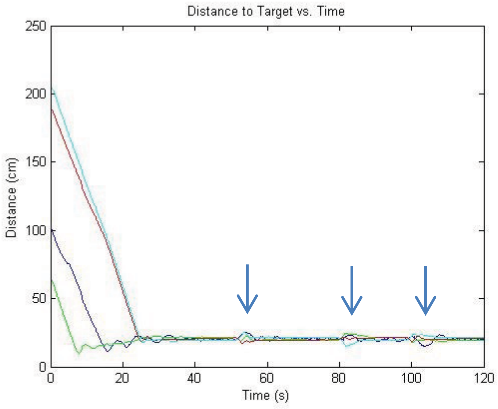


FIGURE 3. SIMULATED DISTANCE BETWEEN AGENTS AND TARGET VS. TIME

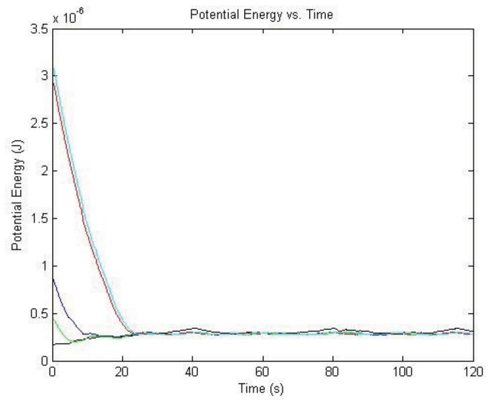


FIGURE 4. SIMULATED AGENT MORSE POTENTIAL ENERGY

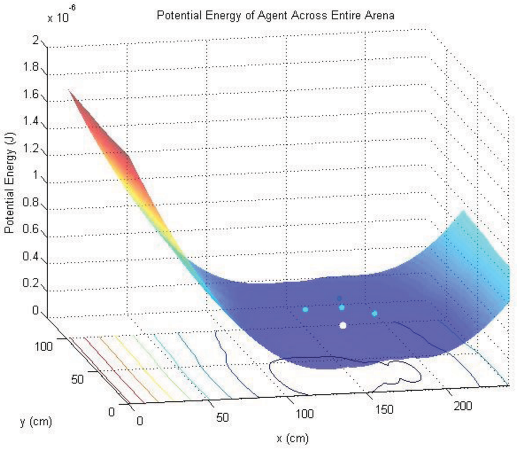


FIGURE 5. POTENTIAL ENERGY OF AGENT ACROSS THE ENTIRE ARENA

In the second simulation, agents identified obstacles and navigated around them due to the obstacle repulsive force. Agents periodically got stuck in local minima and would not advance. Since the system was dynamic, agents moved out of the local minima when the rest of the system changed positions and exerted different forces on the stuck agent.

The third simulation demonstrated the system's ability to accommodate for failure. When an agent stopped moving, the remaining agents continued along their normal trajectories. However, agents redistributed themselves to create a triangular formation around the target as shown in Fig. 6, rather than the previous square formation. The triangular formation minimized gaps between the robots in order to compensate for the missing agent.

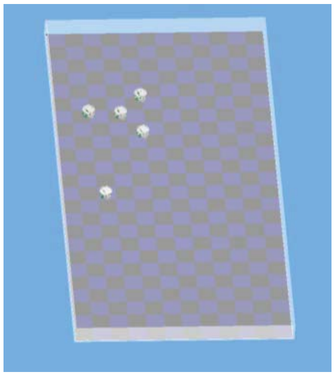


FIGURE 6. SIMULATED FORMATION WITH ONE AGENT FAILURE

EXPERIMENTAL SETUP

The experiment was conducted in the Robotics and Manufacturing Automation (RAMA) Laboratory at Duke University. As seen in Fig. 7, the arena consisted of a flat testbed measuring 120cm by 240cm. Reflective markers were uniquely fastened to each of the five e-puck robots in the arena. Three OptiTrack V100:R2 cameras (not visible in Fig. 7) were mounted above the arena and identified the markers. A computer using the NaturalPoint Tracking Tools software then processed the camera images. Through the program, the spatial locations of the e-pucks were identified and streamed over the network. Thus, the OptiTrack system served as a surrogate GPS system that could provide positioning information to each of the e-pucks. By using the OptiTrack system in the experiment, the assumption was made that all agents had previously been identified and that the system was calibrated correctly.

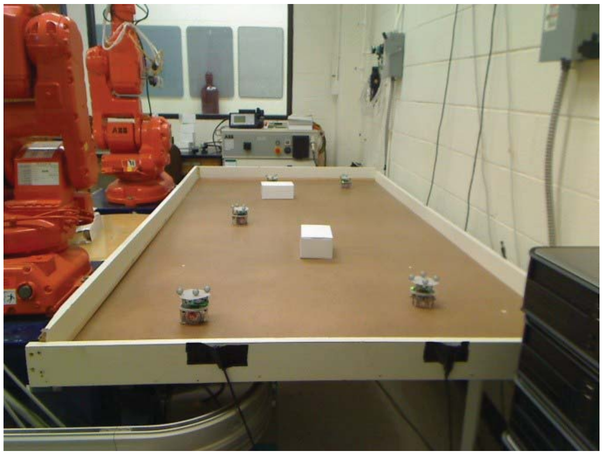


FIGURE 7. DUKE RAMA LAB ARENA

Since the e-puck robots have little processing power, all calculations were performed by an offboard computer. Five computers running MATLAB executed the computations during the experiment; each computer controlling one e-puck.

The computers communicated with their designated e-puck through a bluetooth connection. While the calculations were not performed on board the e-puck, it is important to note that this is not a centralized control system. All computations were performed individually and paths were calculated independently.

Proximity sensor information was transmitted from each e-puck to its respective computer during each control loop. Calibration tests were regularly performed before each experiment to increase proximity sensor consistency. After the proximity sensor and Morse potential calculations were made, individual wheel speeds were communicated to the agent, who in turn actuated them.

At the beginning of the experiment, the five e-pucks were randomly placed across the arena. The e-pucks started with no initial velocity. After establishing connection to their respective computers, the controllers were simultaneously triggered by a clap. Each closed-loop controller ran until terminated by a user-defined timer.

One e-puck was designated as the target during the experiment. To model random motion, its controller consisted of a MATLAB program that randomly generated a set of coordinates in the arena. The target then moved towards those coordinates and was assigned a new destination upon arrival. The remaining agents used a decentralized controller based on the model.

EXPERIMENTAL RESULTS

Plots for the entrapment/escorting mission experiments in an obstacle-free arena are shown in Fig. 8 and Fig. 9. They correspond to the plots generated during the simulation. The experiment validates the simulation's calculation that agents will maintain a distance approximately 20 cm apart from each other, and the Morse potential energy will approach a value of 0.3×10^{-6} J. The equilibrium formation of the agents in an obstacle free arena is shown in an overhead photograph in Fig. 10.

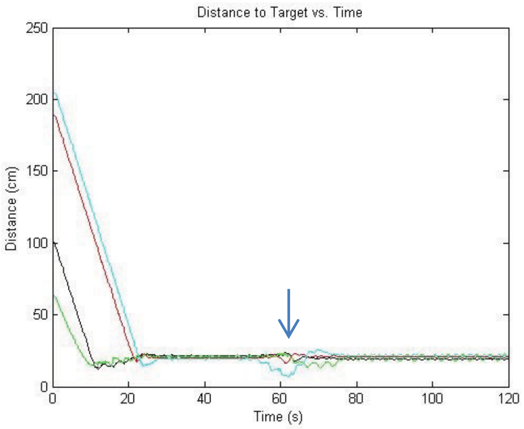


FIGURE 8. EXPERIMENTAL DISTANCE BETWEEN AGENTS AND TARGET VS. TIME

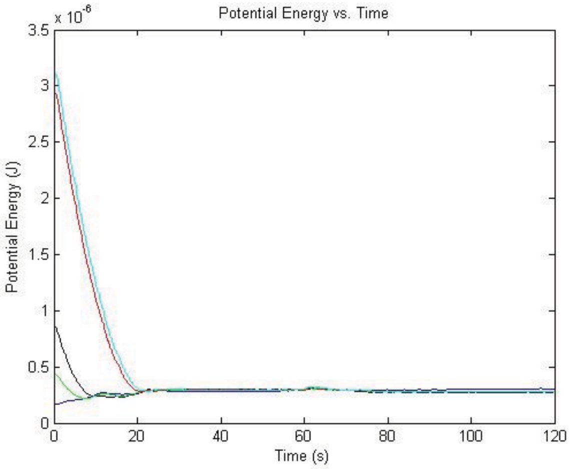


FIGURE 9. EXPERIMENTAL AGENT MORSE POTENTIAL ENERGY

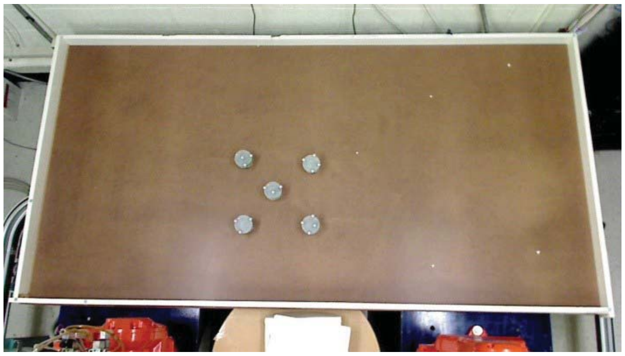


FIGURE 10. EQUILIBRIUM FORMATION

To prevent controllers from setting dangerously high wheel speeds, a velocity cap was placed on each wheel. This limit can be visibly seen in Fig. 8, when agents begin their movement to the target at a constant slope. As indicated by the arrow in Fig. 8, agents temporarily broke their formation as the target moved close to the boundary, forcing agents to move away from the arena's walls to avoid collisions.

Additional testing found that agents were able to successfully redistribute themselves into a triangularly shaped formation when an agent intentionally failed. The method proposed for the e-puck sensors was successful. However, it should be noted that all obstacles in the arena were glossy and white. Since the surfaces were able to strongly reflect the signals from the proximity sensors, their performance was expected to be better than a dark obstacle that would have absorbed the infrared light from the proximity sensors.

CONCLUSION

This paper models, simulates, and experimentally validates that the Morse potential function can be successfully used to exert decentralized control during the entrapment/escorting mission. Agents were able to follow a moving target and evenly distribute themselves, even when a robot failed. Experimental testing successfully demonstrated the application of the model to nonholonomic vehicles. The inclusion of proximity sensors in the algorithm as a detection switch, rather than providing a detection measurement, enabled agents to navigate around obstacles in the arena. Future research applying the Morse potential function to the entrapment/escorting mission might investigate its application in 3-dimensional environments. Additionally, the experiment performed in this paper could be reevaluated using a robot with higher quality sensing.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the United States Army Research Office under grant number W911NF-08-1-0106.

REFERENCES

[1] Arkin, R., 2007. "Robot Ethics," *Research Horizons*, Winter/Spring, pp.14-15.

[2] Kiva Systems [Online]. Available: http://www.kivasystems.com.

[3] Kumar, V., Shanin, F., 2003. "Cognitive Maps in Swarm Robots for the Mine Detection Application". *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3364-3369.

[4] Trifa, V.M., Cianci, C.M., and Guinard, D., 2008. "Dynamic Control of a Robotic Swarm using a Service-Oriented Architecture". *13th International Symposium on Artificial Life and Robotics*.

[5] Young, R.F., 2011. "Modeling, Simulation, and Characterization of Distributed Multi-Agent Systems". *Ph.D. dissertation,* Dept. Mech. Eng. & Mater. Sci., Duke Univ., Durham, NC.

[6] Antonelli, G., Arrichiello, F., and Chiaverini, S., 2008. "The Entrapment/Escorting Mission". *IEEE Robot, Automag, Mag.* 15(1), March, pp.22-29.

[7] Antonelli, G., Arrichiello, F., and Chiaverini, S., 2007. "The Entrapment/Escorting Mission for a Multi-Robot System: Theory and Experiments". *IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* pp.1-6.

[8] Mas, I., Li, S., Acain, J., Kitts, C., 2009. "Entrapment/Escorting and Patrolling Missions in Multi-Robot Cluster Space Control". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.5855-5861.

[9] Rampinelli, V.T.L., Brandão, A.S., Martins, F.N., Sarcinelli-Filho, M., and Carelli, R., 2009. "A Multi-Layer Control Scheme for Multi-Robot Formations with Obstacle Avoidance". *International Conference on Advanced Robotics*, pp. 1-6.

[10] Chen, F., Ren, W., and Cao, Y., 2010. "Surrounding Control in Cooperative Agent Networks". *Systems & Controls Letters*, 59(11), pp.704-712.

[11] Brandão, A.S., Sarapura, J.A., Caldiera, E.M.O., Sarcinelli-Filho, M., and Carelli, R., 2010. "Decentralized Control of a Formation Involving a Miniature Helicopter and a Team of Ground Robots Based on Artificial Vision". *2010 Latin American Robotics Symposium and Intelligent Robotic Meeting*, pp. 126-131.

[12] Kawakami, H., Namerikawa, T., 2009. "Cooperative Target-capturing Strategy for Multi-vehicle Systems with Dynamic Network Topology". *2009 American Control Conference*, pp. 635-640.

[13] Kobayashi, Y., Otsubo, K., and Hosoe, S., 2006. "Design of Decentralized Capturing Behavior by Multiple Mobile Robots". *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pp. 13-18.

[14] Laskowski, D., 2010. "Asset Protection in a Limited Swarm Environment Utilizing Artificial Potential Fields". *Master's Thesis*, Dept. Comp. Eng., Rochester Institute of Technology, Rochester, NY.

[15] LaRocque, J., 2010. "Regional Target Surveillance with Cooperative Robots using APFs". *Master's Thesis*, Dept. Comp. Eng., Rochester Institute of Technology, Rochester, NY.

[16] Gazi, V., Passino, K., 2003. "Stability Analysis of Swarms," *IEEE Transactions on Automatic Control*, 48(4), April, pp.692-697.

[17] Milutinović, D., Lima, P., 2007. "Cells and Robots: Modeling and Control of Large-size Agent Populations," *Springer*.

[18] Liu, W., Short, M.B., Bertozzi, A.L., and Taima, Y.E., 2010. "Multiscale Collaborative Searching through Swarming". *Proc. 7th Int. Conf. On Inform. Control, Autom., Robot.*.

[19] D'Orsonga, M.R., Chuang, Y.O., Bertozzi, A.L., and Chayes, L.S., 2006. "Self-Propelled Particles with Soft-Core Interactions: Patterns, Stability, and Collapse". *Phys. Rev. Lett.,* 96(10), March.

[20] Mabrouk, M.H., 2011. "Individual-Based Model to Simulate Crown Dynamics Using Artificial Potential Fields". *World Congress on Engineering*.

[21] Mondada, F., 2009. "The e-puck, a Robot Designed for Education in Engineering". *Proc. Of the 9th Conference on Autonomous Robot Systems and Competitions*, pp. 59-65.